# EXHIBIT 24

**IN THE UNITED STATES DISTRICT COURT**
**FOR THE WESTERN DISTRICT OF TEXAS**
**WACO DIVISION**

| | | | |
|---|---|---|---|
| Sonos, Inc., | | § | |
| | | § | |
| | Plaintiff, | § | No. 6:20-cv-881-ADA |
| v. | | § | |
| | | § | |
| Google LLC, | | § | |
| | | § | |
| | Defendant. | § | |

**DECLARATION OF DOUGLAS C. SCHMIDT**

I, Douglas C. Schmidt, hereby declare as follows:

## I. SCOPE OF ASSIGNMENT

1. I have been retained by Plaintiff Sonos, Inc. to provide my opinions on how a person of ordinary skill in the art ("POSITA") at the time of the inventions of U.S. Patent Nos. 9,967,615 ("'615 Patent") and 10,779,033 ("'033 Patent") would have understood the following claim terms:

| Patent | Term |
|---|---|
| '033 Patent | "data network" |
| '615 Patent | "local area network" |
| '615 Patent | "a media particular playback system" |

2. This Declaration explains my analysis and opinions of the above-identified claim terms that are used in the '615 or '033 Patent. In forming my opinions, I have read and understand the claims of the '615 and '033 Patents, the specification that is common to both the '615 and '033 Patents, and each of the patents' respective file histories.

3. I reserve the right to supplement or clarify the opinions set forth herein, and if I am requested to do so, to provide additional opinions regarding the '615 and/or '033 Patents.

4. I am being compensated at my normal hourly consulting rate of $550/hour for this matter. My compensation does not depend in any way on the nature of my opinions or the outcome of this case.

## II. SUMMARY OF OPINIONS

5. As explained in detail herein, it is my opinion that a POSITA at the time of the inventions of the ’615 and ’033 Patents would have understood the above-identified claim terms as follows:

| Patent | Term | POSITA’s Understanding |
|---|---|---|
| ’033 Patent | “data network” | “a medium that interconnects devices, enabling them to send digital data packets to and receive digital data packets from each other” |
| ’615 Patent | “local area network” | “data network that interconnects devices within a limited area, such as a home or office” |
| ’615 Patent | “a media particular playback system” | “a media playback system” |

6. I understand that Sonos and/or Google may seek construction of claim terms in the ’615 and/or ’033 Patents other than those expressly addressed herein. I have not analyzed, and express no opinions on, the proper construction of any other claim term in the ’615 or ’033 Patents at this time.

## III. BACKGROUND & QUALIFICATIONS

7. I am the Cornelius Vanderbilt Professor of Engineering in the Department of Electrical Engineering and Computer Science at Vanderbilt University in Nashville, TN, where I also serve as the Associate Provost for Research Development and Technologies and the co-Director of the Data Science Institute. My research spans a broad range of software systems, including distributed object computing, middleware platforms, real-time operating systems, and

distributed real-time and embedded systems. I became a Full Professor with tenure at Vanderbilt University in January 2003.

8. I received my Doctor of Philosophy (Ph.D.) degree in Computer Science from the University of California (UC) Irvine in Irvine, CA in 1994. I also earned a Master's Degree in Computer Science from UC Irvine in 1990, as well as a Bachelor's Degree in Sociology in 1984 and Master's Degree in Sociology in 1986 from the College of William and Mary in Williamsburg, VA. I first started programming in 1983 when I was an undergraduate student taking statistics courses. From 1985 through 1994 I learned how to program in Pascal, C, C++, Ada, Prolog, and Lisp, both at the College of William and Mary and at UC Irvine.

9. I have been a full-time university professor since 1994. I was previously a tenured professor at the University of California, Irvine in the Electrical and Computer Engineering department, from 2000 to 2003, and before that at Washington University in St. Louis, MO in the Computer Science and Engineering department and the Mallinckrodt Institute of Radiology, from 1994 to 1999. In addition, I served as the Chief Technology Officer and Deputy Director for the Software Engineering Institute (SEI) at Carnegie Mellon University from 2010 to 2012, where I led the SEI's research, development, and operational efforts related to software engineering and cyber-security.

10. For the past three decades, my research has focused on distributed real-time and embedded (DRE) systems, which has yielded the ACE, Java ACE, TAO, and CIAO middleware frameworks. The millions of lines of object-oriented code in these frameworks provide layers of infrastructure and distribution middleware that simplify the development of concurrent and networked software apps and services. These middleware frameworks constitute some of the most successful examples of software research and development (R&D) ever transitioned from

research to industry, being widely used by thousands of companies and agencies worldwide in many domains, including national defense and homeland security, datacom/telecom, financial services, healthcare, and online gaming.

11. My research on DRE systems has been funded by various organizations, including both federal agencies, such as DARPA, NSF, NASA, NIH, the U.S. Air Force, and the U.S. Navy, as well as leading companies, such as Northrup Grumman, Raytheon, Lockheed-Martin, Boeing, McDonnell-Douglas, General Electric, Siemens Medical Engineering, and Kodak Health Imaging Systems. I have also received other honors and awards, including election to professional organizations, engagements for invited talks, and the 2015 Award for Excellence in Teaching from the Vanderbilt University Department of Electrical Engineering.

12. Besides my academic and research experience, from 2010 to 2014, I served as a member of the United States Air Force Scientific Advisory Board (SAB), where I was the Vice Chair of the SAB's Cyber Situational Awareness study, which conducted a comprehensive review of the U.S. Air Force's tactics, techniques, and procedures related to secure network-centric mission operations. I have also served on the Advisory Board for the U.S. Naval Air Systems Command (NavAir) Future Airborne Capability Environment (FACE) and was a co-lead of a task force on "Published Open Interfaces and Standards" for the U.S. Navy's Open Systems Architecture initiative.

13. For over 30 years, I have conducted and supervised many research projects involving a wide range of software-related topics, including patterns, optimization techniques, and empirical analyses of communication protocol stacks, web servers, and object-oriented middleware frameworks for distributed real-time embedded systems and mobile-/web-based cloud computing applications. I have published 650+ scholarly articles and technical papers, and

I am the co-author/editor of 10+ books or book-length manuscripts on various topics, including software architecture, network programming, object-oriented frameworks, distributed and real-time systems, open-source middleware platforms, and web-/mobile-based cloud computing applications.

14. My work has been cited 42,500+ times across a comprehensive spectrum of high-impact publications, and my current h-index[1] score is 86, which reveals the significant impact of my publications on scholarly literature in the field of computer science. I have also supervised the research of more than 40 PhD and Master's graduate students to date. Together with conducting and publishing my own research, I have served on the editorial board of many journals, including publications by IEEE and the ACM, and I have been a guest editor of many special issue journals based on my research expertise.

15. On top of my research experience, I have decades of hands-on programming experience with a variety of different programming languages. I began programming with C in 1985 and have programmed with object-oriented languages since 1986, when I began to program with C++. I have programmed with Java and other related object-oriented languages since the mid-1990s and early 2000s. Starting in 1991, while at the University of California Irvine, I led the development of one of the first C++ object-oriented frameworks for concurrent and networked middleware and applications (ACE). Starting in 1996, I developed one of the first Java object-oriented frameworks for concurrent and networked middleware and applications (Java ACE).

[1] The h-index is a popular measure of scholarly productivity. The definition of the index is that a scholar with an index of h has published h papers each of which has been cited in other papers at least h times. Thus, the h-index reflects both the number of publications and the number of citations per publication.

16. Since 1990, I have taught more than 2,000 students in dozens of face-to-face courses on network programming to both undergraduate and graduate students at UC Irvine, Washington University St. Louis, and Vanderbilt University. Since 2013, I have taught mobile cloud computing to more than 400,000 students in online courses, including Massive Open Online Courses (MOOCs) on the Coursera platform, which have focused on technologies like mobile app programming with Android, Java, and JavaScript, as well as programming cloud computing platforms using various web services frameworks, such as Spring and Node.js.

17. Together with my regular course offerings, over the past 30 years I have also taught 600+ short-courses and tutorials on many subjects, including: software design patterns, object-oriented and functional programming; systems programming and network programming for UNIX and Windows; multi-threading and synchronization; concurrent and parallel programming; and various courses on distributed systems, real-time and embedded systems, TCP/IP, web apps and services, compiler construction, algorithms, and data structures.

18. My complete qualifications and professional experiences are described in my curriculum vitae, provided as Appendix A.

## IV. LEGAL STANDARDS

19. I am not an attorney, but I have been informed by counsel about legal standards relevant to my opinions.

20. I understand that claim construction begins with the language of the claims themselves. Claim terms are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the art ("POSITA") when viewing the claim terms in the context of the entire patent.

21. I understand that, in some cases, the plain and ordinary meaning of a claim term may be readily apparent and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words.

22. I understand that, in other cases, a claim term may have a specialized meaning in which case it is often necessary to look to the intrinsic evidence—which I understand to include the claims, the specification, and the prosecution/file history of the patent at issue—to construe the claim term. Indeed, I understand that the context in which a term is used in a claim can be highly instructive. I also understand that the specification is highly relevant to claim construction and can be the single best guide in determining the meaning of a claim term. In this respect, I understand that a claim construction that stays true to the claim language and most naturally aligns with the specification will be the correct construction. Accordingly, I understand that I must refrain from importing limitations into the claims that are not required by the intrinsic evidence.

23. Moreover, I understand that extrinsic evidence – dictionaries, treatises, and the like – can also be used to assist with claim construction. However, I understand that intrinsic evidence is often more reliable than the extrinsic evidence.

## V. LEVEL OF ORDINARY SKILL IN THE ART

24. I have been asked to offer my opinion regarding the level of ordinary skill in the art with respect to the ’615 and ’033 Patents.

25. To assess the level of ordinary skill in the art, I understand one considers the type of problems encountered in the art, the prior solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the level of education of active workers in the field.

26. To assess the level of ordinary skill in the art of the ’615 and ’033 Patents here, I have reviewed the ’615 and ’033 Patents and related documents and considered the type of problems encountered in the art, the prior solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the level of education of active workers in the field. In addition, I considered my own experience teaching and performing research in the networking and consumer audio systems fields, as well as my experience collaborating and consulting with concerns in these industries.

27. Based on my assessment and my personal knowledge and experience in the fields of networking and consumer audio systems, including the configuration and/or control of networked devices, it is my opinion that a person of ordinary skill in the art for purposes of the ’615 and ’033 Patents is a person having the equivalent of a four-year degree from an accredited institution (typically denoted as a B.S. degree) in computer science, computer engineering, electrical engineering, or an equivalent thereof, and approximately 2-4 years of professional experience in the fields of networking and network-based systems or applications, such as consumer audio systems, or an equivalent level of skill, knowledge, and experience.

28. Moreover, based on my education, training, and professional experience discussed in my CV and background, I am very familiar with the level of knowledge and abilities of a person of ordinary skill in the art at the time of the inventions of the ’615 and ’033 Patents.

29. In forming the opinions set forth herein, I applied the level of ordinary skill in the art set forth above.

## VI. OVERVIEW OF THE ’615 & ’033 PATENTS

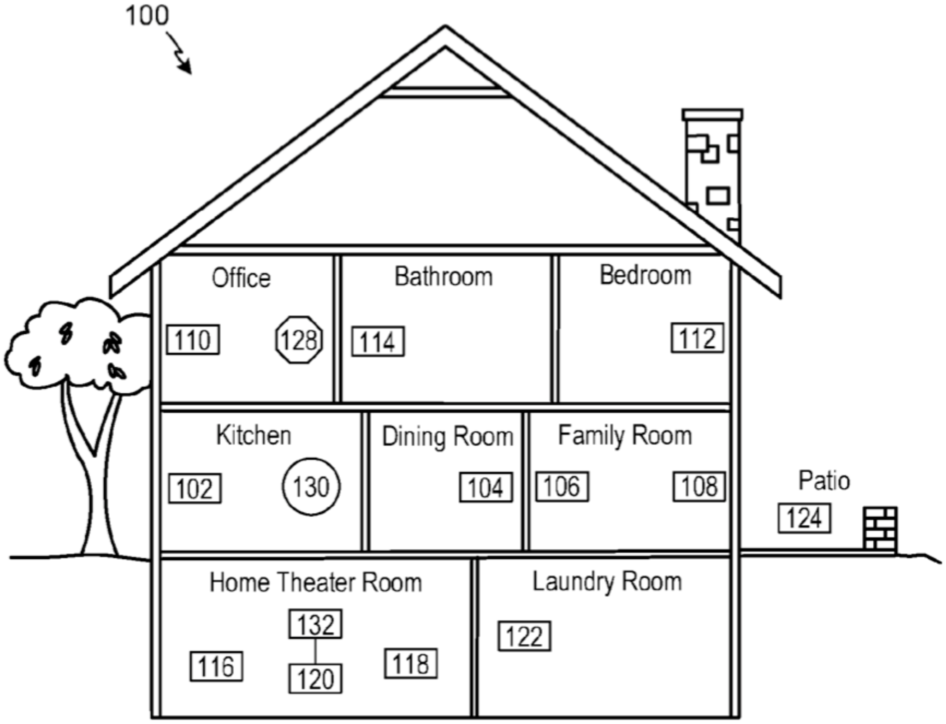
30. The ’615 and ’033 Patents are part of the same patent family and stem from the same original patent application, application number 13/341,237 (the “’237 Application”), filed

by Sonos on December 30, 2011. In particular, the ’615 Patent was filed on February 23, 2015 and is a direct continuation of the ’237 Application. The ’033 Patent was filed on April 19, 2019, is part of a different branch in this patent family than the ’615 Patent, and ultimately claims priority back to the ’237 Application through a sequence of continuations. For this Declaration, I have been asked to assume that the invention date for the ’615 and ’033 Patents is December 30, 2011.

31. The ’615 and ’033 Patents share a common specification. Thus, for consistency, my citations in this Declaration to the disclosures in this common specification are with reference to the column and line numbers of the ’615 Patent’s specification. That said, it should be understood that the same teachings are also found in the ’033 Patent’s specification.

32. The ’615 Patent describes a “local playback system” (sometimes referred to as a “home music system” or “household playback system”) comprising one or more “playback devices” (also referred to as “zone players”) that connect to a local “data network” (also referred to as a “local area network”) and are capable of playing back multimedia content, such as audio. *See, e.g.*, ’615 Patent at 1:13-15, 1:66-2:9, 2:51-3:13, 3:28-31, 5:21-54, 10:64-66, 12:44-67, 16:1-8. The ’615 Patent further describes control devices (e.g., “network-enabled portable devices,” such as smart phones) that also connect to the local “data network” and are capable of controlling the operation of the “local playback system” (such a control device is sometimes referred to as a “controller”). *See, e.g., id.* at 3:18-37, 4:52-5:11.

33. Figure 1 of the ’615 Patent provides an illustrative example of a “local playback system” at a user’s home comprising a variety of “playback devices” 102-124, a control device 130, and a “data network 128”:




*See, e.g.*, *id.* at 3:18-37, 5:21-28. It is my opinion that a POSITA would understand that "data network 128" represents a local area network (LAN). *See, e.g.*, *id.* at 10:64-66, 16:1-8.

34. The '615 Patent also provides an illustrative example of a local "data network" that takes the form of an "Ad-Hoc network 610" and is communicatively coupled to a "cloud-based" "data network" (e.g., the Internet):

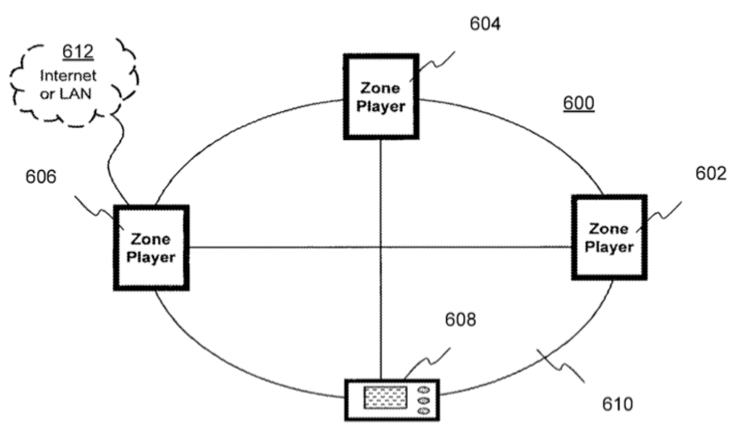



FIGURE 6

*See, e.g.*, *id.* at 10:56-12:3.

35. In the disclosed "local playback system," each "playback device" is capable of communicating over the local "data network" with various other networked devices, including one or more other "playback devices," one or more control devices, and one or more local audio sources. *See, e.g.*, '615 Patent at 4:40-52, 6:61-7:12, 7:37-66, 8:12-16, 10:66-11:9, FIGS. 1, 6. Likewise, each "playback device" and control device is capable of communicating over a wide-area network (e.g., via the local "data network"), such as to retrieve audio from an Internet-based audio source. *See, e.g.*, '615 Patent at 5:38-41, 6:64-7:12, 12:44-67, FIG. 6.

36. The '615 Patent further provides an illustrative example of a system architecture including a cloud-based "data network" (e.g., the Internet) and multiple "local playback systems" on respective local "data networks" (760, 770):

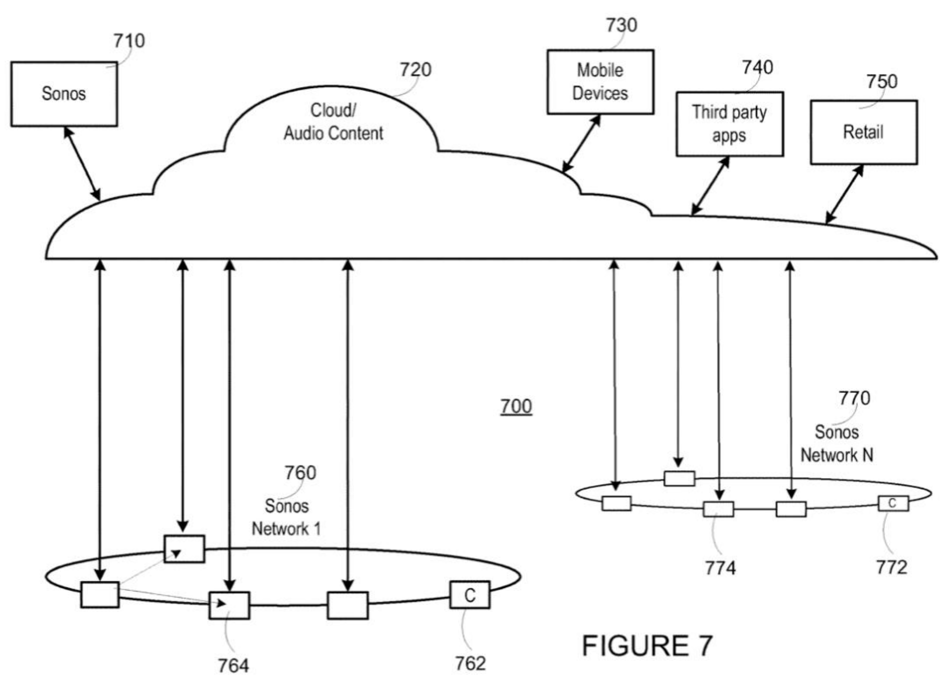


FIGURE 7

*See, e.g.*, '615 Patent at 12:19-43, 16:1-8.

37. The '615 Patent explains that the communications over the local and cloud "data networks" are in the form of digital data "packets" and are in accordance with one or more

standard communication protocols, such as IEEE 802.11a, 802.11b, 802.11g, 802.11n, 802.15, or 802.3. *See, e.g.*, '615 Patent at 7:37-66, 11:45-51.

38. As disclosed in the '615 Patent, control devices and "playback devices" may communicate with one another over a cloud-based "data network" to facilitate transferring playback from one device to another. For instance, the '615 Patent discloses a variety of situations where a user is listening on his/her personal computing device to music from an Internet-based, music application (e.g., Pandora, Rhapsody, Spotify, etc.) and decides to instead have that playback be transferred to one or more "playback devices" in his/her "local playback system." *See, e.g.*, '615 Patent at 12:44-13:30. The example cloud-based system architecture illustrated in Figure 7 of the '615 Patent enables the user's personal computing device to communicate with one or more cloud-based servers to facilitate the transfer of playback from the personal computing device to one or more "playback devices" in a "local playback system." *See, e.g.*, *id.* at 12:19-43, 15:18-46, 16:1-8, 17:12-20.

## VII. "DATA NETWORK"

39. The first term that I was asked to analyze is "data network," which is found in the independent claims of the '033 Patent. I understand that Sonos and Google have offered the following constructions for this term:

| Sonos's Proposed Construction | Google's Proposed Construction |
|---|---|
| "a medium that interconnects devices, enabling them to send digital data packets to and receive digital data packets from each other" | Plain and ordinary meaning; no construction necessary at this time |

40. It is my opinion that Sonos's proposed construction is consistent with how a POSITA would have interpreted the term "data network" in the context of the '033 Patent (and '615 Patent) because it appropriately specifies that a "data network" (i) is a medium that

interconnects devices, enabling the devices to both send and receive information (i.e., it enables two-way communication) and (ii) transfers information in the form of digital data packets. A POSITA would appreciate that these are fundamental characteristics of a "data network," which was (and still is) a term commonly used in the field of networking.

41. While it is unclear why Google disagrees with Sonos's construction, it is my opinion that a POSITA at the time of the invention would not have interpreted the term "data network" in the context of the '033 Patent (and '615 Patent) different from Sonos's proposed construction. In this respect, a POSITA would have understood at the time of the invention that, for example, a one-way communication link, such as an infrared remote sending a signal to a TV, would not amount to a "data network." Likewise, as another example, a POSITA would have understood at the time of the invention that speaker wire connecting an AVR to passive speakers to carry analog audio signals would not amount to a "data network."

42. My opinions in this respect are based on my analysis of the intrinsic evidence of the '033 Patent (and '615 Patent) and determination of how a POSITA would have understood the meaning of the term "data network" in the context of the '033 Patent (and '615 Patent) at the time of the invention. My opinions are also based on my review of extrinsic evidence regarding the term "data network" (and analogous terms), including dictionary definitions and other technical sources. Below, I set forth an explanation of the bases of my opinions with respect to this term.

### A. "Data Network" Is Synonymous With Other Well-Understood Terms

43. For starters, it is my opinion that a POSITA would understand from reading the independent claims of the '033 Patent and the specification that the term "data network" is used

in a synonymous manner as other well-understood terms in the field of networking, such as "data communications network," "computer network," and "packet network."

44. In this respect, the independent claims of the '033 Patent expressly recite that the claimed "computing device" (e.g., the control device discussed above) is "communicatively coupled . . . over a data network" to one or more "playback devices in a media playback system[.]" The independent claims further make clear to a POSITA that the claimed "computing device" and "playback devices" have computer capabilities, such as the capabilities of receiving and processing data from an Internet-based source. *See, e.g.*, ''033 Patent at claim 1:

> "A computing device comprising . . .
>
> program instructions stored on the non-transitory computer-readable medium that, when executed by the at least one processor, cause the computing device to perform functions comprising: . . .
>
> operating in a first mode in which the computing device is configured for playback of a remote playback queue provided by a cloud-based computing system associated with a cloud-based media service; . . .
>
> based on receiving the user input, transmitting an instruction for the at least one given playback device to take over responsibility for playback of the remote playback queue from the computing device, wherein the instruction configures the at least one given playback device to (i) communicate with the cloud-based computing system in order to obtain data identifying a next one or more media items that are in the remote playback queue, (ii) use the obtained data to retrieve at least one media item in the remote playback queue from the cloud-based media service; and (iii) play back the retrieved at least one media item; . . . .

45. Accordingly, a POSITA would understand that the claimed "data network" is being used in a similar manner as the well-understood term "computer network" was often used in the field of networking around the time of the invention.

46. The teachings of the '033 Patent further confirm that the term "data network" is being used by the '033 Patent in an synonymous manner as other common networking terms, such as "data communications network," "computer network," and "packet network," were used

around the time of the invention. For instance, the ’033 Patent explains that “zone players” (or “playback devices”) and control devices are coupled to “data network 128” either wirelessly or via a wire (e.g., via an Ethernet cable). *See, e.g.*, ’615 Patent at 4:64-5:3, 5:21-23. The ’033 Patent explains that the “data network 128” may utilize “mesh network” and/or Ethernet technology, which a POSITA would understand are computer-network technologies. *See, e.g.*, ’615 Patent at 5:29-41:

> Particularly, the data network 128 can be a wired network, a wireless network, or a combination of both. In some embodiments, one or more of the zone players 102-124 are wirelessly ***coupled to the data network 128 based on a proprietary mesh network***. In some embodiments, one or more of the zone players 102-124 are wirelessly coupled to the data network 128 using a non-mesh topology. In some embodiments, one or more of the zone players 102-124 are ***coupled via a wire to the data network 128 using Ethernet or similar technology***. In addition to the one or more zone players 102-124 connecting to the data network 128, the data network 128 can further allow access to a wide area network, such as the Internet.[2]

47. The ’033 Patent also explains that the “data network 128” provides “zone players” with “access to a wide area network, such as the Internet,” which a POSITA would understand is a common function of a computer network. *See, e.g.*, ’615 Patent at 5:38-41.

48. Further still, the ’033 Patent explains that the “data network 128” can be created by connecting a “zone player” to a “broadband router” that in turn “can be connected to an Internet Service Provider” and “can be used to form another data network within the system configuration 100, which can be used in other applications (e.g., web surfing).” *Id.* at 5:41-54. This is yet further confirmation for a POSITA that “data network” in the context of this patent is used in a similar manner as “data communications network,” “computer network,” and “packet network” were often used around the time of the invention. Other teachings from the ’033 Patent demonstrating this synonymous use of these terms are discussed below.

[2] All emphasis added herein unless otherwise specified.

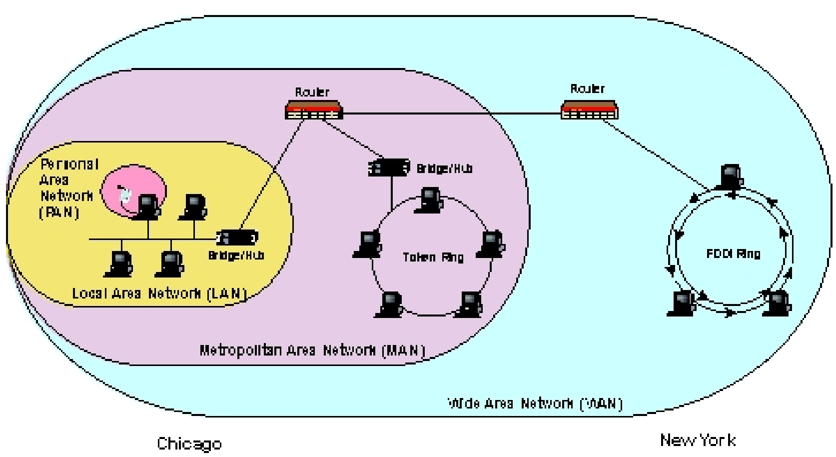
49. My opinion that a POSITA would understand that the ’033 Patent uses the term “data network” in a similar manner as well-understood terms like “data communications network,” “computer network,” and “packet network” is supported by a variety of technical sources.

50. As a first example, *VOIP Industry Dictionary* (2009) defines “data network” as follows:

> A data network is a system that **transfers data** between network access points (nodes) through data switching, system control and interconnection transmission lines. Data networks are primarily ***designed to transfer data from one point to one or more points*** (multipoint). Data networks may be composed of a variety of communication systems including circuit switches, leased lines and packet switching networks.

SONOS-SVG2-00018823 at 23 (bold without italics original) (attached as Appendix B).

51. The online *VOIP Industry Dictionary* provides the below figure illustrating different types of “data networks” containing a variety of ***computers*** transferring data and explains that the coverage range of the different types of “data networks” can vary from relatively short (e.g., personal area networks (PANs) and local area networks (LANs)) to relatively wide (e.g., “wide area data networks (WANs)”).




Data Network Type Diagram

SONOS-SVG2-00018823 at 23.

52. Thus, the *VOIP Industry Dictionary* makes clear that the term "data network" refers to a network of computers and one common type of a "data network" is a "local area network" (LAN).

53. As another example, *Data Communications and Computer Networks for Computer Scientists and Engineers* (2nd ed. 2003) defines a "local area network" as "[a] ***data communications network*** used to interconnect a community of digital devices over a localized area . . . ." SONOS-SVG2-00018417 at 22 (attached as Appendix C).

54. In view of this definition and that of the *VOIP Industry Dictionary*, it is clear that a POSITA would understand that a "local area network" is one type of "data network" that is also referred to as a "data communications network."

55. As yet another example, *Local & Metropolitan Area Networks* (6th ed. 2000) explains that "LANs . . . and WANs are all ***examples of communications networks***," where a "communications network" is "a facility that interconnects a number of devices and provides a means for ***transmitting data*** from one attached device to another." SONOS-SVG2-00018752 at 57 (attached as Appendix D). *Local & Metropolitan Area Networks* later explains that "LANs" are often "distinguished from ***other types of data networks***" based on their use in a limited geographic area. *Id*. at 59.

56. As before, this technical source makes clear that a POSITA would understand that a "local area network" is one type of "data network" that is also referred to as a "data communications network."

57. As a further example, *Computer Networks And Internets* (2nd ed. 1999) provides a brief history of "computer networking" and LANs:

> The history of ***computer networking*** changed dramatically during the late 1960s and early 1970s when researchers developed a form of ***computer communication*** known as Local Area Networks (LANs). . . . Each LAN consists of a single shared medium, usually a cable, to which many computers attach. The ***computers*** take turns using the medium to send packets.

SONOS-SVG2-00018301 at 5 (attached as Appendix E). *Computer Networks And Internets* also explains that "[LAN] technologies have become the most popular form of ***computer networks***" and that, as of its writing, "LANs now connect more computers than any other type of network." *Id.*

58. In view of this technical source and the above-cited sources, it is clear that a POSITA would understand that a "local area network" is one type of "computer network" that is also referred to as a "data communications network" or "data network."

59. As yet other examples, *Webster's New World Telecom Dictionary* (2008) states that "[a] LAN is a ***packet network*** designed to interconnect host computers, peripherals, storage devices, and other computing resources within a local area, i.e., limited distance," (SONOS-SVG2-00018832 at 36 (attached as Appendix F), and *Packet Broadband Network Handbook* (2004) states that (i) "[a] local area network is a high-speed ***data network*** that covers a relatively small geographic area" and (ii) a "LAN is a type of broadband ***packet*** access ***network*** that carries the ***packet data*** traffic of an organization." SONOS-SVG2-00018673 at 76 (attached as Appendix G).

60. In view of these technical sources and the above-cited sources, it is clear that a POSITA would understand that a "local area network" is one type of "packet network" that is also referred to as a "data network," "data communications network," or "computer network."

61. Thus, in my opinion, there is no doubt that the term "data network" is used in the '033 Patent in a similar manner as the well-understood terms "packet network," "data

communications network" and "computer network" were often used around the time of the invention.

62. As I alluded to above, a POSITA would have understood at the time of the inventions that these terms "data network," "packet network," "data communications network," and "computer network" each refer to a network with at least two fundamental characteristics: (i) a medium that interconnects devices, enabling the devices to both send and receive information (i.e., two-way communication) and (ii) information is exchanged in the form of digital data packets.

### B. Medium Enabling Two-Way Communication of Information

63. Consistent with the well-understood meanings of the networking terms "data network," "packet network," "data communications network," and "computer network" from around the time of the inventions, the '033 Patent repeatedly and consistently explains that the disclosed "data networks" must enable two-way communication between connected devices. In my opinion, this repeated and consistent disclosure would lead a POSITA to understand that the term "data network" in the context of the '033 Patent refers to a medium that interconnects devices that enables them to send information to and receive information from each other.

64. For instance, the '033 Patent repeatedly and consistently explains that the local "data network 128" enables two-way communications between "playback devices," control devices, and local audio sources.

> [T]he network interface 402 [of a zone player 400] facilitates ***a data flow between zone players and other devices on a data network (e.g., the data network 128 of FIG. 1) and the zone player 400***. In some embodiments, the network interface 402 can manage the assembling of an audio source or file into smaller ***packets that are to be transmitted over the data network*** or ***reassembles received packets*** into the original source or file. In some embodiments, the network interface 402 can further handle the ***address part of each packet so that it gets to the right destination*** or ***intercepts packets destined for the zone player 400***. Accordingly, in certain

> embodiments, each of the packets includes an Internet Protocol (IP)-based source address as well as an IP-based destination address.

’615 Patent at 7:37-50.

> The wireless interface 404 [of the zone player 400], also referred to as an RF interface, provides network interface functions for the zone player 400 to ***wirelessly communicate with other devices (e.g., other zone player(s), speaker(s), receiver(s), component(s) associated with the data network 128***, and so on) in accordance with a communication protocol (e.g., any of the wireless standards IEEE 802.11a, 802.11b, 802.11g, 802.11n, or 802.15). To ***receive wireless signals*** and to provide the wireless signals to the wireless interface 404 and ***to transmit wireless signals***, the zone player 400 of FIG. 4 includes one or more antennas 420. The wired interface 406 provides network interface functions for the zone player 400 to communicate over a wire with other devices in accordance with a communication protocol (e.g., IEEE 802.3).

’615 Patent at 7:53-66.

> The example zone player 204 of FIG. 2C does not include an amplifier, but allows a receiver 214, or another audio and/or video type device with built-in amplification, to ***connect to a data network 128 of FIG. 1*** and to play ***audio received over the data network 128*** via the receiver 214 and a set of detached speakers 216. . . . In some embodiments the zone player 202 can ***transmit a second signal to, for example, other zone player(s) in the same or different zone(s), speaker(s), receiver(s), and so on***.

’615 Patent at 4:13-25.

> [A] zone player can ***relay one or more signals received from, for example, a first zone player to another playback device***. In some embodiments, a zone player can receive a first signal and generate an output corresponding to the first signal and, simultaneously or separately, can ***receive a second signal and transmit or relay the second signal to another zone player(s), speaker(s), receiver(s), and so on***. Thus, an example zone player described herein can act as a playback device and, at the same time, ***operate as a hub in a network of zone players***. In such instances, media content corresponding to the first signal can be different from the media content corresponding to the second signal.

’615 Patent at 4:40-52; *id.* at 6:61-7:1 (“Sources of audio content to be played by zone players 102-124 are numerous. ***Music*** from a personal library stored on a computer or networked-attached storage (NAS) ***can be accessed via the data network 128*** and played.”), 7:5-8 (“***Audio content received*** from one or more sources ***can be shared*** amongst the zone players 102 to 124

***via the data network 128*** and/or the controller 130."); *see also, e.g.*, *id.* at 3:56-64, 4:64-65, 7:12-18, 8:12-16.

65. Likewise, the '033 Patent explains that "Ad-hoc network 610" -- which a POSITA would understand, in the context of the '033 Patent, is one specific example of a local "data network" -- enables two-way communications between "zone players" 602-606 and a "controller 608." *See e.g.*, '615 Patent at 10:66-11:2 ("With an established Ad-Hoc network 610, the devices 602, 604, 606 and 608 can ***all communicate with each other in a 'peer-to-peer' style of communication***, for example."), 11:6-9 ("Using the Ad-Hoc network 610, the devices 602, 604, 606, and 608 ***can share or exchange*** one or more audio sources and be grouped to play the same or different audio sources.").

66. Further yet, the '033 Patent repeatedly and consistently explains that "playback devices," control devices, and cloud servers are able to perform two-way communications with one another utilizing one or both of a local and/or cloud-based "data network."

> Sources of audio content to be played by zone players 102-124 are numerous. . . . ***Internet*** radio stations, shows, and podcasts ***can be accessed via the data network 128***. ***Music services*** that let a user ***stream and download*** music and audio content ***can be accessed via the data network 128***.

'615 Patent at 6:61-7:1.

> The zone player 606 in FIG. 6 is shown to be ***connected to both networks***, for example. The connectivity to the network 612 is based on Ethernet while the connectivity to other devices 602, 604 and 608 is based on Wireless. It is understood, however, that in some embodiments each zone player 606, 604, 602 may ***access the Internet when retrieving media from the cloud (e.g., Internet)*** via the bridging device. For example, zone player 602 may contain a uniform resource locator (URL) that specifies ***an address to a particular audio track in the cloud***. Using the URL, the zone player 602 may ***retrieve the audio track from the cloud***, and ultimately play the audio out of one or more zone players.

'615 Patent at 11:65-12:3.

> ***Using the cloud 710***, a multimedia playback system 720 (e.g., Sonos™), a mobile device 730, a third party application 740, a retail location 750, and so on can ***provide multimedia content (requested or otherwise) to local playback networks*** 760, 770. ***Within each local network*** 760, 770, a controller 762, 772 and/or playback device 764, 774 can ***provide*** a song identifier, song name, playlist identifier, playlist name, genre, preference, and so on, ***and/or simply receive*** content from a connected system ***via the cloud***.

'615 Patent at 12:34-43.

> For example, a user listens to a third party music application (e.g., Pandora™ Rhapsody™, Spotify™, and so on) on her smart phone while commuting. She's enjoying the current channel and, as she walks in the door to her home, selects an option to continue playing that channel on her household music playback system (e.g., Sonos™). The playback system picks up from the same spot on the selected channel that was on her phone and outputs that content (e.g., that song) on speakers and/or other playback devices connected to the household playback system. A uniform resource indicator (URI) (e.g., a uniform resource locator (URL)) can be ***passed to a playback device*** to ***fetch content from a cloud and/or other networked source***, for example. A playback device, such as a zone player, can ***fetch content on its own*** without use of a controller, for example. Once the zone player has a URL (or some other identification or address) for a song and/or playlist, the zone player can run on its own to ***fetch the content***. Songs and/or other multimedia content can be ***retrieved from the Internet*** rather than a local device (e.g., a compact disc (CD)), for example.

'615 Patent at 12:44-63.

> In certain embodiments, information is provided from a third party application to a local playback system without being routed through or by a controller application. Here, the third party application is ***communicating with*** the multimedia playback device (e.g., a Sonos ZonePlayer™). Information can be ***passed locally, rather than through the Internet***, for example. The local playback device ***accesses the Internet to find content to stream***, and the third party application takes the place of the controller application . . .

'615 Patent at 16:9-17.

> [T]he third party application not only tells the local playback system what to play, but also ***maintains two-way communication*** with the local playback (e.g., Sonos™) system. ***Two-way communication*** helps enable features such as keeping a local playback queue synchronized with a queue that the user is editing/managing in the third party application; allow the third party application to know what is currently playing on the local playback system; allow integrated transport control between the third party application and the local playback system; and so on.

’615 Patent at 16:18-31.

> Certain embodiments facilitate control of a local playback system ***from outside a household or other location at which the local playback network*** is configured. For example, a user can queue up music while away from his or her house. The application can facilitate setup and/or configuration. For example, a third party application may ask the user to enter a Sonos customer email address and password. The application can then make a ***request to a Sonos server in the cloud*** to determine the zone groups on which music can be played.

’615 Patent at 17:12-20.

67. The ’033 Patent’s repeated and consistent explanation that its disclosed “data networks” must enable two-way communication between connected devices is directly consistent with the well-understood meaning of the analogous terms “data network,” “packet network,” “data communications network,” and “computer network,” as evidenced by a variety of technical sources.

68. For example, *Local & Metropolitan Area Networks* (6th ed. 2000) states that “[a] communications network is a facility that ***interconnects*** a number of devices and provides a ***means for transmitting data from one*** attached device ***to another***.” SONOS-SVG2-00018752 at 67. It also explains that the IEEE 802 committee refers to a “LAN” as a “data network” and defines a LAN as “a ***shared medium peer-to-peer communications network*** that broadcasts information to all stations to receive.” *Id*. at 59. It further explains that each device that attaches to a LAN typically has a “network interface card” that contains logic for not only accessing the LAN but also “***for sending and receiving*** blocks of data on the LAN.” *Id*.

69. As another example, *Data & Computer Communications* (6th ed. 2000) explains, in its “data communication networking” section’s introduction, that the “two major categories into which communications networks are traditionally classified” are “wide area networks (WANs) and local area networks (LANs).” SONOS-SVG2-00018715 at 18 (attached as

Appendix H). It goes on to explain that, "[a]s with WANs, a LAN is a communication network that ***interconnects*** a variety of devices and provides ***a means for information exchange*** among those devices" and that, "[a]t each station, there is a ***transmitter/receiver*** that communicates over ***a medium shared*** by other stations." *Id.* at 21. It provides the following summary of a LAN: "[a] LAN consists of a shared ***transmission medium*** and a set of hardware and software for interfacing devices to the medium and regulating the orderly access to the medium." *Id*. at 23.

70. As yet another example, *Encyclopedia of Networking* (1998) explains that a LAN "is a connectionless networking scheme, meaning that once a workstation is ready to ***transmit*** and has access to the ***shared medium***, it simply puts the packets on the network and hopes that the recipient ***receives*** them" and explains that "[d]ata is packaged into frames for ***transmission*** on the LAN" and "[a] frame is usually ***addressed for a single computer***, although a ***multicast address*** can be used to ***transmit to all workstations*** on the LAN." SONOS-SVG2-00018707 at 10 (attached as Appendix ).

71. As a further example, *Packet Broadband Network Handbook* (2004) explains that "common applications of LAN include . . . ***file exchange between*** connected users" and "LAN is a type of broadband packet access network that ***carries the packet data traffic*** of an organization. LAN ***interconnects*** the end users of an organization to an outside public data network such as the Internet." SONOS-SVG2-00018673 at 76.

72. I note that it is my understanding that the well-understood meaning of the terms "data network," "packet network," "data communications network," and "computer network" has not materially changed from the 2000 timeframe up to today.

73. Thus, both the intrinsic and extrinsic evidence support my opinion that a POSITA would understand that the term "data network" in the context of the '033 Patent refers to a

medium that interconnects devices that enables them to send information to and receive information from each other.

### C. Information in the Form of Digital Data Packets

74. Also consistent with the well-understood meaning of the networking terms "data network," "packet network," "data communications network," and "computer network" from the time of the invention, a POSITA would readily understand from the teachings and context of the '033 Patent that the information exchanged over the disclosed "data networks" must take the form of digital data packets (sometimes referred to as "frames").

75. For starters, as I mentioned before, a POSITA reading the claims and disclosure of the '033 Patent with an invention date no later than December 2011 would appreciate that the "data networks" claimed and disclosed by the '033 Patent are in the context of modern computer networks, which a POSITA would understand involve the exchange of information in digital form.

76. In fact, the '033 Patent expressly explains that information exchanged over the disclosed "data networks" takes the form of digital data "packets":

> [T]he network interface 402 facilitates a ***data flow*** between zone players and other devices ***on a data network*** (e.g., the data network 128 of FIG. 1) and the zone player 400. In some embodiments, the network interface 402 can manage the assembling of an audio source or file into smaller ***packets*** that are to be transmitted over the data network or reassembles received ***packets*** into the original source or file. In some embodiments, the network interface 402 can further handle the ***address part of each packet*** so that it gets to the right destination or intercepts ***packets*** destined for the zone player 400. Accordingly, in certain embodiments, each of the ***packets*** includes an Internet Protocol (IP)-based source address as well as an IP-based destination address.

'615 Patent at 7:37-50.

77. What's more, the '033 Patent repeatedly explains that the network transmissions over a "data network" are performed in accordance with communication protocols that a

POSITA would appreciate involved digital data packets at the time of the invention. *See, e.g.*, '615 Patent at 7:48-50 ("[E]ach of the packets includes an ***Internet Protocol (IP)-based*** source address as well as an ***IP-based*** destination address."), 7:53-66:

> The wireless interface 404 . . . provides network interface functions for the zone player 400 to wirelessly communicate with other devices (e.g., other zone player(s), speaker(s), receiver(s), component(s) associated with the data network 128, and so on) ***in accordance with a communication protocol*** (e.g., any of the wireless standards ***IEEE 802.11a, 802.11b, 802.11g, 802.11n, or 802.15***). To receive wireless signals and to provide the wireless signals to the wireless interface 404 and to transmit wireless signals, the zone player 400 of FIG. 4 includes one or more antennas 420. The wired interface 406 provides network interface functions for the zone player 400 to communicate over a wire with other devices ***in accordance with a communication protocol*** (e.g., ***IEEE 802.3***).

11:45-51 ("In certain embodiments, configuration of a HOUSEHOLD involves multiple CPs and ZPs that rendezvous and establish a known configuration such that they can use a standard networking protocol (e.g., ***IP*** over Wired or Wireless ***Ethernet***) for communication. In an embodiment, two types of networks/***protocols*** are employed: ***Ethernet 802.3*** and Wireless ***802.11g***.").

78. A POSITA's understanding that the "data networks" disclosed in the '033 Patent handle exchanging information in the form of digital data packets is also supported by the well-understood meaning of the analogous terms "data network," "packet network," "data communications network," and "computer network," as evidenced by a variety of technical sources.

79. For example, as discussed before, several technical sources state that a "LAN" or other "data network" is a "***packet*** network" or "***packet*** access network" that carries information in digital form. *See, e.g.*, SONOS-SVG2-00018673 at 76 ("LAN is a type of broadband packet access network . . . . The physical layer [of the LAN protocols] is primarily concerned with the

***transmission medium*** and its physical characteristics for ***digital signal transmission***.”); SONOS-SVG2-00018832 at 36 (“A LAN is a packet network . . . .”).

80. As another example, *Computer Networks And Internets* (2nd ed. 1999) explains that “[e]ach LAN consists of a single shared medium” and “computers take turns using the medium to send ***packets***.” SONOS-SVG2-00018301 at 5. It goes on to explain that “computers take turns using the medium to send data” and “[a]lthough ***LAN technologies require computers to divide data into small packets called frames***, only one ***packet*** can be transmitted on a LAN at any time.” *Id.* at 12.

81. As yet another example, *Data Communications and Computer Networks for Computer Scientists and Engineers* (2nd ed. 2003) defines a LAN as “[a] data communications network used to interconnect a community of ***digital devices*** over a localized area” and explains that “[m]essages within a LAN are transmitted as a series of variable length ***frames*** using transmission media which introduce only relatively low error rates.” SONOS-SVG2-00018417 at 20, 22. It also provides some history explaining that “[m]any modern LANs evolved from a LAN known as Aloha which was . . . ***packet*** based and used radio as its transmission medium.” *Id.* at 21.

82. As a further example, *Encyclopedia of Networking* (1998) explains that, in a LAN, “once a workstation is ready to transmit and has access to the shared medium, it simply puts the ***packets*** on the network and hopes that the recipient receives them.” SONOS-SVG2-00018707 at 10. It provides the following explanation of how information is packaged:

> ***Data*** is packaged into ***frames*** for transmission on the LAN. . . . A ***frame*** is usually ***addressed*** for a single computer, although a multicast ***address*** can be used to transmit to all workstations on the LAN. Higher-layer protocols such as ***IP*** and IPX package ***data into datagrams***. ***Datagrams*** are in turn divided up and put into ***frames*** for transmission on a particular LAN.

*Id.* at 10, 11.

83. As yet another example, *Data & Computer Communications* (6th ed. 2000) explains that WANs (a common type of a "data network") have been implemented using "packet switching" and "ATM" technologies. SONOS-SVG2-00018715 at 18. With respect to "packet switching" technology, it explains: "***data*** are ***sent out*** in a sequence of ***small chunks, called packets***. . . . Packet-switching networks are commonly used for terminal-to-computer and computer-to-computer communications." *Id.* at 19. *Data & Computer Communications* also explains that "packet switching was developed at a time when ***digital*** long-distance transmission facilities exhibited a relatively high error rate compared to today's facilities." *Id.* And with respect to "ATM" technology, it explains:

> Asynchronous transfer mode (ATM), sometimes referred to as cell relay, is a culmination of all of the developments in circuit switching and packet switching over the past 25 years. ATM can be viewed as an evolution from frame relay. The most obvious difference between frame relay and ATM is that frame relay uses ***variable-length packets, called frames***, and ATM uses ***fixed-length packets, called cells***.

*Id.* at 20.

84. Thus, both the intrinsic and extrinsic evidence support my opinion that a POSITA would understand that the term "data network" in the context of the '033 Patent involves a type of network that facilitates exchanging information in the form of digital data packets.

### D. Conclusion Regarding "Data Network"

85. As explained above, both the intrinsic and extrinsic evidence support my opinion that Sonos's proposed construction is consistent with how a POSITA would have interpreted the term "data network" in the context of the '033 Patent (and '615 Patent). This is because both the intrinsic and extrinsic evidence demonstrate that a POSITA would understand that each disclosed "data network" in the '033 Patent (i) is a medium that interconnects devices, enabling the devices

to both send and receive information and (ii) transfers information in the form of digital data packets.

## VIII. "LOCAL AREA NETWORK"

86. The next term that I was asked to analyze is "local area network" (LAN), which is found in the independent claims of the '615 Patent. I understand that Sonos and Google have offered the following constructions for this term:

| Sonos's Proposed Construction | Google's Proposed Construction |
|---|---|
| "data network that interconnects devices within a limited area, such as a home or office" | Plain and ordinary meaning; no construction necessary at this time |

87. It is my opinion that Sonos's proposed construction is consistent with how a POSITA would have interpreted the term "local area network" in the context of the '615 Patent (and '033 Patent) because it appropriately specifies that a "local area network" (i) is a "***data network***" as opposed to just ***any*** type of "network," and (ii) interconnects devices within a ***limited*** geographic area. A POSITA would appreciate that these are fundamental characteristics of a "local area network," which was (and still is) a term commonly used in the field of networking.

88. While it is unclear why Google disagrees with Sonos's construction, it is my opinion that a POSITA at the time of the invention would not have interpreted the term "local area network" in the context of the '615 Patent (and '033 Patent) different from Sonos's proposed construction. In this respect, a POSITA would have understood at the time of the invention that, for instance, neither an infrared remote sending a signal to a TV nor speaker wire connecting an AVR to passive speakers would amount to a "local area network." Likewise, a POSITA would have understood at the time of the invention that two devices communicatively

coupled to one another ***only*** by way of an ***Internet*** connection would not be communicatively coupled by way of a "local area network."

89. My opinions in this respect are based on my analysis of the intrinsic evidence of the '615 Patent (and '033 Patent) and determination of how a POSITA would have understood the meaning of the term "local area network" in the context of the '615 Patent (and '033 Patent) at the time of the invention. My opinions are also based on my review of extrinsic evidence regarding the term "local area network," including dictionary definitions and other technical sources. Below, I set forth an explanation of the bases of my opinions with respect to this term.

### A. A "Local Area Network" Is a "Data Network"

90. As explained in detail above, both the intrinsic and extrinsic evidence confirm that a POSITA would understand that the '615 Patent uses the term "local area network" to refer to one type of a "data network," which as I explained in detail above has the following meaning in the context of the '615 Patent: a medium that interconnects devices, enabling them to send digital data packets to and receive digital data packets from each other.

91. Additional extrinsic evidence further confirms that a POSITA would understand that the '615 Patent's "local area network" includes the features of a "data network." *See, e.g.*, SONOS-SVG2-00018838 at 38 [*Google.com* (retrieved Apr. 7, 2021)] (defining "local area network" as "a ***computer network*** . . . .") (attached as Appendix J); SONOS-SVG2-00018237 at 39-40 [*A Guide to Computer Network Security* (2009)] ("A ***computer network*** with two or more computers or clusters of network and their resources connected by a ***communication medium*** sharing communication protocols and confined in a small geographical area . . . is called a local area network (LAN).") (attached as Appendix K); SONOS-SVG2-00018402 at 6 [*Encyclopedia of Computer Science and Technology* (2009)] (attached as Appendix L):

> There are two basic ways to connect computers in a LAN. The first [is] called Ethernet . . . . Ethernet uses a single cable line called a bus to which all participating computers are connected. Each ***data packet*** is received by all computers, but processed only by the one it is addressed to. . . . Naturally there must be software to manage the ***transmission and reception*** of ***data packets***. The structure of a ***packet*** (***sometimes called a frame***) has been standardized . . . .

SONOS-SVG2-00018832 at 36 [*Webster's New World Telecom Dictionary* (2008)] ("A LAN is a ***packet network*** designed to interconnect host computers, peripherals, storage devices, and other computing resources within a local area, i.e., limited distance."); SONOS-SVG2-00018673 at 76 [*Packet Broadband Network Handbook* (2004)]:

> A local area network is a high-speed ***data network*** that covers a relatively small geographic area. . . . LAN is a type of broadband ***packet*** access ***network*** that carries the ***packet data*** traffic of an organization. LAN interconnects the end users of an organization to an outside public data network such as the Internet. . . . The physical layer [of the LAN protocols] is primarily concerned with the ***transmission medium*** and its physical characteristics for ***digital signal transmission***."

SONOS-SVG2-00018670 at 72 [*Newton's Telecom Dictionary* (2003)] ("A short distance ***data communications network*** (typically within a building or campus) used to link computers and peripheral devices . . . under some form of standard control.") (attached as Appendix M).

### B. <u>A LAN Interconnects Devices in a Limited Area</u>

92. Consistent with the well-understood meaning of "local area network" in the field of networking, the '615 Patent makes clear to a POSITA that it uses the term "local area network" to refer to a "data network" that interconnects devices within a ***limited*** area, such as a home or office.

93. In fact, the lead embodiment illustrated in Figure 1 of the '615 Patent shows "data network 128," which a POSITA would understand is a "local area network," interconnecting devices within a single home. *See, e.g.*, '615 Patent at FIG. 1; *see also, e.g., id.* at 5:12-28, 12:44-49 ("For example, a user listens to a third party music application (e.g., Pandora™

Rhapsody™, Spotify™, and so on) on her smart phone while commuting. She's enjoying the current channel and, as she walks in the door to her ***home***, selects an option to continue playing that channel on her ***household*** music playback system (e.g., Sonos™).").

94. Moreover, the '615 Patent refers to a "local area network" as a "***small***" network. *See, e.g.*, *id.* at 10:64-66 ("In general, an Ad-Hoc (or 'spontaneous') network is a ***local area network*** or ***other small network*** in which there is no one access point for all traffic.").

95. What's more, the '615 Patent's repeated and consistent use of the term "local" to contrast with terms connoting large geographic areas (or wide coverage ranges), such as "wide area network," "cloud," "remote," and "Internet," would lead a POSITA to the inescapable conclusion that the term "local area network" in the '615 Patent refers to a "data network" that interconnects devices within a ***limited*** area. *See, e.g.*, '615 Patent at 5:38-41 ("In addition to the one or more zone players 102-124 connecting to the ***data network 128*** [that interconnects devices of a local playback system within a home], the ***data network 128*** can further allow access to ***a wide area network***, such as the ***Internet***."), 12:19-21 ("FIG. 7 shows a system including a plurality of networks including a ***cloud-based network*** and at least one ***local*** playback network."), 12:31-38:

> As illustrated by the example system 700 of FIG. 7, a plurality of content providers 720-750 can be connected to one or more ***local*** playback networks 760-770 via a ***cloud*** and/or other network 710. Using the ***cloud*** 710, a multimedia playback system 720 (e.g., Sonos™), a mobile device 730, a third party application 740, a retail location 750, and so on can provide multimedia content (requested or otherwise) to ***local*** playback networks 760, 770.

12:53-56 ("A uniform resource indicator (URI) (e.g., a uniform resource locator (URL)) can be passed to a playback device to fetch content from a ***cloud*** and/or other networked source, for example. . . . Songs and/or other multi- media content can be retrieved from the ***Internet rather than*** a ***local*** device (e.g., a compact disc (CD)), for example."), 16:1-8:

> A connection between the third-party application and the ***local*** playback device (e.g., Sonos ZonePlayer™) can be direct over a ***local area network (LAN)***, ***remote*** through a proxy server in the ***cloud***, and so on. A ***LAN*** delivery approach may be easier to integrate into 'native' applications (e.g., applications written for iOS or Android), and a ***proxy server*** approach may be easier for third party applications that are browser-based, for example.

16:13-15 ("Information can be passed ***locally***, ***rather than through the Internet***, for example."),

17:12-20:

> Certain embodiments facilitate control of a ***local*** playback system from ***outside a household*** or other location at which the ***local*** playback network is configured. For example, a user can queue up music while ***away from*** his or her ***house***. The application can facilitate setup and/or configuration. For example, a third party application may ask the user to enter a Sonos customer email address and password. The application can then make a request to a Sonos ***server in the cloud*** to determine the zone groups on which music can be played.

*See also, e.g., id.* at 13:1-22, 13:36-40, 13:60-14:28, 14:42-43 15:18-32, 15:38-44.

96. The '615 Patent's use of the term "local area network" to refer to a "data network" that interconnects devices within a ***limited*** area is not surprising given that this is consistent with the well-understood meaning of "local area network" in the field of networking, as evidenced by a variety of technical sources. Some examples of which include:

- SONOS-SVG2-00018237 at 39, 40 [*A Guide to Computer Network Security* (2009)] ("A computer network with two or more computers or clusters of network and their resources connected by a communication medium sharing communication protocols and ***confined*** in a ***small geographical area***, such as a building floor, a building, or a few adjacent buildings, is called a local area network (LAN).")

- SONOS-SVG2-00018832 at 36 [*Webster's New World Telecom Dictionary* (2008)] ("A LAN is a packet network designed to interconnect host computers, peripherals, storage devices, and other computing resources within a ***local area***, ***i.e., limited distance***.")

- SONOS-SVG2-00018673 at 76 [*Packet Broadband Network Handbook* (2004)] ("A local area network is a high-speed data network that covers a relatively ***small geographic area***.")

- SONOS-SVG2-00018417 at 20 [Duck & Reed, *Data Communications and Computer Networks for Computer Scientists and Engineers*, 2nd Edition (2003)] ("LANs generally encompass a ***small physical area*** . . . and are usually ***confined within a single site***.")

- SONOS-SVG2-00018301 at 12 [Douglas E. Comer, *Computer Networks And Internets*, 2nd Edition (1999)] ("Designed for use over a ***small distance (e.g., in a building)***, a LAN does not need a separate wire between each pair of computers.")

97. As with the terms "data network," "packet network," "data communications network," and "computer network," I note that the meaning of the term "local area network" has not materially changed from the 2000 timeframe up to today.

98. Thus, both the intrinsic and extrinsic evidence support my opinion that a POSITA would understand that the term "local area network" in the context of the '615 Patent refers to a "data network" that interconnects devices within a ***limited*** area, such as a home or office.

## IX. "A MEDIA PARTICULAR PLAYBACK SYSTEM"

99. The final term that I analyzed is "a media particular playback system," which is found in dependent claims 3, 15, and 26 of the '615 Patent. I understand that Sonos contends that the word "particular" was erroneously included in this phrase and thus, this phrase should instead read "a media playback system." On the other hand, I understand that Google contends that this phrase is indefinite.

100. In my opinion, a POSITA having read any of dependent claims 3, 15, or 26 would readily understand that the word "particular" was erroneously included in the phrase "a media particular playback system" and that the phrase should instead read "a media playback system," as Sonos contends.

101. Each of dependent claims 3, 15, and 26 recites the same additional limitations to its respective independent claim. I have reproduced the claim language for each of these claims below with the phrase at issue highlighted in red.

3. The method of claim 1, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

15. The tangible, non-transitory computer readable medium of claim 13, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

26. The control device of claim 25, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

102. In my opinion, it is evident from the face of the ’615 Patent to a POSITA that the phrase “a media particular playback system” contains a typographical error -- namely, the

inadvertent inclusion of the word "particular." Indeed, in reading the claims in their entirety, a POSITA would recognize that the specific sequence of words in the phrase "a media particular playback" is unnatural and not how a POSITA would purposefully write in this context.

103. Moreover, other claims of the '615 Patent would confirm to a POSITA that the inclusion of the word "particular" in the phrase "a media particular playback system" was a typographical error.

104. For example, as shown below, dependent claims 2 and 3 each have similar structure but claim 2 includes the more natural phrase "a media playback system" (highlighted in green), whereas claim 3 includes the phrase "a media particular playback system":

| 2. The method of claim 1, | 3. The method of claim 1, |
|---|---|
| wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone of a media playback system that includes the particular playback device as a first channel of a stereo pair and an additional playback device as a second channel of the stereo pair, | wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, |
| wherein modifying the one or more transport controls of the control interface to control playback by the particular playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the additional playback device, and | wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and |
| wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the additional playback device playing back the multimedia content as the stereo pair. | wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony. |

105. As another example, as shown below, dependent claims 14 and 15 each have similar structure but claim 14 includes the more natural phrase "a media playback system" (highlighted in green), whereas claim 15 includes the phrase "a media particular playback system":

| | |
|---|---|
| 14. The tangible, non-transitory computer readable medium of claim 13, | 15. The tangible, non-transitory computer readable medium of claim 13, |
| wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone of a media playback system that includes the particular playback device as a first channel of a stereo pair and an additional playback device as a second channel of the stereo pair, | wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, |
| wherein modifying the one or more transport controls of the control interface to control playback by the particular playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the additional playback device, and | wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and |
| wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the additional playback device playing back the multimedia content as the stereo pair. | wherein the particular playback device playing back the retrieved multimedia content comprises the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony. |

106. Given the parallelisms between claims 2 and 3 and claims 14 and 15, it is my opinion that a POSITA would understand that Sonos intended for (i) claim 3 to recite "a media playback system," as recited in claim 2, and (ii) claim 15 to recite "a media playback system," as recited in claim 14.

107. In addition to the above-discussed claim language, the specification of the '615 Patent confirms that the only reasonable correction to the erroneous phrase "a media particular playback system" is to remove the word "particular."

108. For starters, the inventions of the '615 Patent are generally described in the context of a "local playback system" that includes "one or more multimedia playback devices," which a POSITA would understand is an example of what the phrase "media playback system" refers to. *See, e.g.*, '615 Patent at 2:51-57, 2:60-3:1, 12:44-67.

109. Moreover, neither the phrase "media particular" nor "multimedia particular" is found anywhere in the '615 Patent other than in claims 3, 15, and 26. On the other hand, the phrases "media playback" or "multimedia playback" (without the intervening word "particular") can be found throughout the '615 Patent. *See, e.g.*, Abstract, 1:66-2:1, 2:51-57, 2:60-3:1, 3:5-13, 15:51-57, 16:11-13, 16:35-40, claims 2, 14. In my opinion, this would lead a POSITA to understand that the phrase "a media particular playback system" was a typographical error.

110. Further still, I have seen nothing in the prosecution history of the '615 Patent that would suggest to a POSITA that any other reasonable correction to the phrase "a media particular playback system" would be appropriate. In fact, the prosecution history confirms that the only reasonable correction is to remove the word "particular."

111. For instance, on October 25, 2016, Sonos amended then-pending independent claim 1, in relevant part, as follows (with underlining indicating additions and strikethroughs indicating deletions):

> causing, via a control device, a graphical interface to display a control interface including one or more transport controls to control playback by the control device;
> <u>after connecting to a local area network via a network interface, identifying, via the control device, playback devices connected to the local area network;</u>
> <u>causing, via the control device, the graphical interface to display a selectable option for transferring playback from the control device;</u>

detecting, via the control device, a set of inputs to transfer playback from the control device to a particular playback device, wherein ~~detecting~~ the set of inputs comprises: (i) a selection of the selectable option for transferring playback from the control device and (ii) a selection of the particular playback device from the identified playback devices connected to the local area network . . . .

Oct. 25, 2016 Office Action Response at p. 2 (underling and strikethroughs original) (attached as Appendix N). Sonos amended the other independent claims in a similar manner. *See id.* at pp. 6, 19. As I have highlighted in teal above, Sonos introduced the adjective "particular" before "playback device."

112. To maintain proper antecedent basis and consistency throughout the claims, Sonos also propagated the adjective "particular" before "playback device" to the dependent claims, including dependent claims 3, 15, and 26. However, as shown below for dependent claim 3, Sonos inadvertently inserted the adjective "particular" before the word "playback" contained in the larger, original phrase "a media playback system" (highlighted in red):

3. The method of claim 1, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

Oct. 25, 2016 Office Action Response at pp. 3-4 (underling and strikethroughs original). This same error was also propagated to the dependent claims that ultimately issued as dependent claim 15 and 26, each of which recites the same additional limitations as claim 3. *See id.* at pp. 7-8, 11.

113. In view of the foregoing, it is my opinion that a POSITA would understand that (i) the phrase "a media particular playback system" found in dependent claims 3, 15, and 26 contains an obvious error, (ii) the face of the '615 Patent makes clear that the only reasonable correction for this error is to remove the word "particular" from the phrase, and (iii) there is nothing in the prosecution history suggesting any other reasonable correction should apply.

I declare under penalty of perjury that the foregoing is true and correct to the best of my knowledge and belief.

Douglas C. Schmidt

Dated: April 27, 2021

DOUGLAS C. SCHMIDT

# Appendix A

# Dr. Douglas Craig Schmidt

Cornelius Vanderbilt Professor of Engineering
Department of Electrical Engineering & Computer Science
Vanderbilt University
Nashville, TN 37203

douglas.c.schmidt@vanderbilt.edu
(TEL) 615-294-9573
(FAX) 615-343-7440
(WEB) www.dre.vanderbilt.edu/∼schmidt/

## Educational Background

- **Ph.D. Computer Science**, summer 1994, University of California, Irvine
  Dissertation: "An Object-Oriented Framework for Experimenting with Alternative Process Architectures for Parallelizing Communication Subsystems."
  Co-advisors: Dr. Tatsuya Suda and Dr. Richard W. Selby.
- **M.S. Computer Science**, summer 1990, University of California, Irvine, specializing in software engineering.
- **M.A. Sociology**, summer 1986, College of William and Mary, Williamsburg, Virginia
  Thesis: "A Statistical Analysis of University Resource Allocation Policies."
  Advisor: Dr. Michael A. Faia.
- **B.A. Sociology**, summer 1984, College of William and Mary, Williamsburg, Virginia.

## Professional Experience

1. **7/1/18 – present: Associate Provost of Research Development and Technologies**
   Develop cohesive and sustainable information technology (IT) services to advance research and scholarship across Vanderbilt's ten schools and colleges; develop scalable storage and processing solutions by leveraging on-campus and cloud data storage services, as well as creating big data research cores and core-related services; and implement NIST 800-171 compliant IT services.
2. **8/1/18 – present: Co-Director of the Vanderbilt Data Science Institute**
   Facilitate highly innovative research and education initiatives that build on Vanderbilt University's current strengths, promote new collaborations, and establish a cohesive institutional framework that embraces Vanderbilt's diverse campus, while establishing the university as a leader in data science research and education.
3. **2/17 – present: Cornelius Vanderbilt Professor of Engineering**
   Received an endowed chair in recognition of my scholarship, intellect, and leadership in the field of computer science and computer engineering.
4. **1/03 – present: Full Professor with tenure**
   Conducting research on patterns, optimizations, and experimental analysis of advanced generative software techniques that facilitate the development of distributed real-time and embedded middleware and model driven architectures running over high-speed networks and interconnects in the Department of Electrical Engineering and Computer Science at Vanderbilt University.
5. **02/16 – 7/31/18: Associate Chair of Electrical Engineering and Computer Science**
   Provide intellectual leadership within the EECS department. Coordinate with EECS Chair to assist in EE, CS, and CompE curriculum development and course staffing. Assist the faculty in building industry and federal programs for EECS. Assist the Chair in mentoring junior EECS faculty. Assist the EECS Chair in improving the ranking of the EECS programs. Assist the Chair in increasing the quality and number of undergraduate and graduate student applications to the EECS programs.
6. **12/04 – 1/16: Associate Chair of Computer Science and Engineering**
   Provide intellectual leadership within the CS program. Coordinate with EECS Chair to assist in CS and CompE (CS&E) curriculum development and course staffing. Assist the faculty in building industry and federal programs centered in CS&E and IT for EECS. Assist the Chair in mentoring

junior CS&E faculty. Assist the EECS Chair in improving the ranking of the CS&E programs. Assist the Chair in increasing the quality and number of undergraduate and graduate student applications to the CS&E programs.

7. **4/13 – 2/18: Member of the Board of Directors at Real-Time Innovations (RTI).** Work with the CEO and other members of the Board of Directors of RTI to help assess company technical and business strategy.

8. **1/12 – present: Visiting Scientist at the Software Engineering Institute**
Assist the SEI Director's Office in formulating the SEI's technology strategy for R&D projects and external relationships by aligning the expertise of the SEI technical staff to identify and respond to the needs of sponsors, customers, and partners and help the SEI shape future innovations in complex software-reliant systems.

9. **7/11 – 7/13: Adjunct Professor of Software Engineering** in the Institute for Software Research in the School of Computer Science at Carnegie Mellon University.

10. **9/10 – 12/11: Deputy Director and Chief Technology Officer at the Software Engineering Institute (SEI)**
Lead the formulation of the SEI's technology strategy for R&D projects and external relationships by aligning the expertise of the SEI technical staff to identify and respond to the needs of sponsors, customers, and partners and help the SEI shape future innovations in complex software-reliant systems.

11. **07/05 – 8/10: Visiting Scientist at the Software Engineering Institute**
Assisted Linda Northrop and the Ultra-Large-Scale (ULS) Systems team to define the challenge problems, promising technology areas, and research roadmaps for the national R&D effort on building the software-reliant systems of the future that are likely to have billions of lines of code. This activity is defining a broad, multi-disciplinary research agenda for developing ULS systems of the future.

12. **06/09 – 8/10: Chief Technology Officer for Zircon Computing**
Assisted in the strategic direction of Zircon Computing technology development in the areas of adaptive distributed computing middleware for high-performance and real-time applications. Help to formulate the technology strategy for open-source middleware platforms, R&D partnerships, and external relationships.

13. **6/07 – 8/07: Visiting Professor at Trinity College Dublin**
Worked with Professor Vinny Cahill and the Distributed Systems Group at Trinity College on topics pertaining to service-oriented architectures and autonomic computing.

14. **10/06 – 5/09: Chief Technology Officer for PrismTechnologies**
Assisted in the strategic direction of PrismTechnologies technology development in the areas of open-source middleware platforms and model-driven tools. Help to formulate the technology strategy for open-source middleware platforms and model-driven tools, R&D partnerships, and external relationships.

15. **3/02 – 12/02: Program Manager**
Led the National effort on middleware as a Program Manager for over $60 million dollars of funding at the DARPA Information Exploitation Office (IXO). Programs include Program Composition for Embedded Systems (PCES) and National Experimentation Platform for Hybrid and Embedded Systems (NEPHEST).

16. **9/01 – 3/02: Deputy Director**
Served as the Deputy Director for the DARPA Information Technology Office (ITO), helping set and guide the National IT research and development agenda and manage programs on autonomous systems, network-centric command and control systems, combat systems, real-time avionics systems, distributed real-time and embedded systems, and augmented cognition for the U.S. Department of Defense.

17. **6/00 – 3/02: Program Manager**
Led the National effort on middleware as a Program Manager for over $60 million dollars of funding at the DARPA Information Technology Office (ITO). Programs included the Program Composition for Embedded Systems (PCES).

18. **6/01 – 6/02: Co-chair for the Software Design and Productivity (SDP) Coordinating Group**
The SDP Coordinating Group formulates the multi-agency research agenda in fundamental software design for the Federal government's Networking and Information Technology Research and Development (NITR&D) Program, which is the collaborative IT research effort of the major Federal science and technology agencies.

19. **8/99 – 2002: Associate Professor with tenure**
Conducted research on patterns, implementation, and experimental analysis of object-oriented techniques that facilitate the development of high-performance, distributed real-time and embedded computing systems on parallel processing platforms running over high-speed networks and embedded system interconnects in the Department of Computer Engineering at the University of California, Irvine.

20. **6/99 – 8/99: Associate Professor with tenure**
Conducted research on patterns, implementation, and experimental analysis of object-oriented techniques that facilitate the development of high-performance, distributed real-time and embedded computing systems on parallel processing platforms running over high-speed networks and embedded system interconnects in the Department of Computer Science and the Department of Radiology at Washington University in St. Louis.

21. **6/98 – 6/99: Associate Professor without tenure (early promotion)**
Conducted research on patterns, implementation, and experimental analysis of object-oriented techniques that facilitate the development of high-performance, distributed real-time and embedded computing systems on parallel processing platforms running over high-speed networks and embedded system interconnects in the Department of Computer Science and the Department of Radiology at Washington University in St. Louis.

22. **8/94 – 6/98: Assistant Professor**
Conducted research on object-oriented patterns and techniques for developing highly extensible, high-performance communication frameworks in the Department of Computer Science and the Department of Radiology at Washington University in St. Louis.

23. **3/91 – 8/94: Research Assistant**
Developed object-oriented frameworks for multi-processor-based communication subsystems with Professor Tatsuya Suda at the University of California, Irvine.

24. **6/90 – 11/90: Member of the Technical Staff**
Worked as a software engineer for Independence Technologies, which was one of the largest suppliers of enterprise-level TUXEDO systems, providers of professional services, and developers of management and connectivity software to support OLTP environments.

25. **8/88 – 3/91: Research Assistant**
Devised measurement-guided software development techniques for large-scale software systems with Professor Richard Selby at the University of California, Irvine.

26. **6/88 – 8/88: Research Assistant**
Studied the impact of computing on end-users in forty U.S. city governments with Dr. John King and the URBIS project at the Public Policy Research Organization, University of California, Irvine.

27. **Summer of 87: Technical Intern**
Worked with Dr. Peter G. W. Keen at the International Center for Information Technology, Washington D.C. on various projects, including software productivity, videotex, and smartcards.

28. **9/86 – 5/88: Teaching Assistant**
Developed programming assignments, grading tools, and led recitation sessions for a number of undergraduate Computer Science courses at the University of California, Irvine.

29. **Summer of 86: Statistical Programmer**
Programmed SPSS and SAS applications for the "Justice Delayed" project under the direction of Dr. Gene Flango at the National Center for State Courts, Williamsburg, Virginia.

30. **1/85 – 8/86: Research Assistant**
Examined university resource allocation policies via statistical analysis under the direction of Dr. Michael Faia at the College of William and Mary, Williamsburg, Virginia.

# Publications

## In Print

- **Refereed Journal Publications**

J127 Alex Roehrs, Cristiano A. da Costa, Rodrigo R. Righi, Andre H. Mayer, Valter F. da Silva, Jose R. Goldim, and Douglas C. Schmidt, "Integrating Multiple Blockchains to Support Distributed Personal Health Records," the SAGE *Health Informatics Journal*, 2021 (to appear).

J126 Zhongwei Teng, Peng Zhang, Xiao Li, William Nock, Denis Gilmore, Marcelino Rodriguez-Cancio, Jules White, Jonathan C. Nesbitt, Douglas C. Schmidt, "Authentication and Integration Approaches for mHealth Apps from a Usability View," the journal *Advances in Electrical and Electronic Engineering*, 2021 (to appear).

J125 Scott Eisele, Aron Laszka, Douglas C. Schmidt, and Abhishek Dubey, "The Role of Blockchains in Multi-Stakeholder Transactive Energy Systems," the journal *Frontiers in Blockchain: Emerging Technologies and Blockchain in Action: Applications in Supply Chain Management and Energy*, to appear 2021.

J124 Peng Zhang, Chris Downs, Nguyen Thanh Uyen Le, Cory Martin, Paul Shoemaker, Clay Wittwer, Luke Mills, Liam Kelly, Stuart Lackey, Douglas C. Schmidt, Jules White, "Towards Patient-centered Stewardship of Research Data and Research Participant Recruitment with Blockchain Technology," the *Frontiers in Blockchain special selection on Non-Financial Blockchain*, 2020, volume 3, pps. 1-32.

J123 Yao Pan, Fangzhou Sun, Jules White, Douglas C. Schmidt, Jacob Staples, Lee Krause, and Zhongwei Teng, "Detecting Web Attacks with End-to-End Deep Learning," the Springer *Journal of Internet Services and Applications*, 2019, volume 10, number 16, pps. 1-22.

J122 Shelagh Mulvaney, Lori Laffel, Korey Hood, Cindy Lybarger, Sarah Vaala, and Douglas C. Schmidt, "A Mobile App Identifies Momentary Psychosocial and Contextual Factors Related to Mealtime Self-Management in Adolescents with Type 1 Diabetes," Journal of the American Medical Informatics Association, Oxford University Press, 2019, Volume 26, Number 12, pps. 1627-1631.

J121 Maria E. Powell, Marcelino Rodriguez Cancio, David Young, William Nock, Beshoy Abdelmessih, Amy Zeller, Irvin Perez Morales, Peng Zhang, C Gaelyn Garrett, Douglas Schmidt, Jules White, and Alexander Gelbard, "Decoding Phonation with Artificial Intelligence (DEP AI): Proof of Concept," the *Laryngoscope Investigative Otolaryngology* journal, Wiley-Blackwell, Volume 4, Issue 3, 2019, pps. 328-334.

J120 Alex Roehrs, Cristiano Andre da Costa, Rodrigo da Rosa Righi, Valter Ferreira da Silva, Jose Roberto Goldim, and Douglas C. Schmidt, "Analyzing the Performance of a Blockchain-based Personal Health Record Implementation," the *Journal of Biomedical Informatics*, Elsevier, volume 92, 2019.

J119 Peng Zhang, Breck Stodghill, Cory Pitt, Cavan Briody, Douglas C. Schmidt, Jules White, Alan Pitt, and Kelly Aldrich, "OpTrak: Tracking Opioid Prescriptions via Distributed Ledger Technology," the *International Journal of Information Systems and Social Change (IJISSC)*, Special Issue On: Blockchain Technology: Platforms, Tools, and Use Cases, IGI Global, Volume 10, Number 2, 2019.

J118 Peng Zhang, Jules White, Douglas C. Schmidt, Gunther Lenz, S. Trent Rosenbloom, "FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data," the Elsevier *Computational and Structural Biotechnology Journal* – Blockchain and Distributed Ledger Technologies in Biology, Medicine, and eHealth Special Issue, Volume 16, July 2018, pp 267–278.

J117 Shelagh A Mulvaney, Sarah Vaala, Korey K Hood, Cindy Lybarger, Rachel Carroll, Laura Williams, Douglas C Schmidt, Kevin Johnson, Mary S Dietrich, and Lori Laffel, "Mobile Momentary Assessment and Bio-Behavioral Feedback for Adolescents with Type 1 Diabetes: Feasibility, Engagement Patterns, and Relation with Blood Glucose Monitoring," ¡EM¿Journal of Diabetes Technology and Therapeutics¡/EM¿, Vol 20, No. 7, July 2018, pp 465–474.

J116 Subhav Pradhan, Abhishek Dubey, Shweta Khare, Saideep Nannapaneni, Aniruddha Gokhale, Sankaran Mahadevan, Douglas C Schmidt, Martin Lehofer, "CHARIOT: A Holistic, Goal Driven Orchestration Solution for Resilient IoT Applications," the ACM Transactions on Cyber-Physical Systems, Vol 2, No. 3, July 2018, pp 1-37.

J115 Hull PC, Emerson JS, Quirk ME, Canedo JR, Jones JL, Vylegzhanina V, Schmidt D, Mulvaney S, Beech B, Husaini BH, "A Smartphone App for Families With Preschool-Aged Children in a Public Nutrition Program: Prototype Development and Beta-Testing," *Journal of Medical Internet Research (JMIR): mHealth and uHealth*, Vol 5, No. 8, August, 2017, pp 1–19.

J114 Yao Pan, Jules White, Douglas C. Schmidt, Ahmed Elhabashy, Logan Sturm, Jaime Camelio, and Christopher Williams, "Taxonomies for Reasoning About Cyber-physical Attacks in IoT-based Manufacturing Systems," Special Issue on Advances and Applications in the Internet of Things, edited by Vicente Garcia Diaz, International Journal of Interactive Multimedia and Artificial Intelligence, volume 4, number 3, 2017, pp. 45-54.

J113 Gordon Blair, Douglas C. Schmidt, and Chantal Taconet, "Middleware for Internet Distribution in the Context of Cloud Computing and the Internet of Things," Springer Journal Annals of Telecommunications, April 2016, Volume 71, Issue 3, pp. 87-92.

J112 Yu Sun, Jules White, Sean Eade, and Douglas C. Schmidt, "ROAR: A QoS-Oriented Modeling Framework for Automated Cloud Resource Allocation and Optimization", the Journal of Systems and Software, Elsevier, volume 116, issue C, June 2016 pp. 146.161.

J111 Nick Guertin, Brian Womble, Paul Bruhns, Douglas C. Schmidt, Adam Porter, and Bill Antypas, "Management Strategies for Software Infrastructure in Large-Scale Cyber-Physical Systems for the US Navy," Cutter IT Journal, Vol. 28, No. 5, May 2015, pp. 14-18.

J110 Jules White, Josi A. Galindo, Tripti Saxena, Brian Dougherty, David Benavides, Douglas C. Schmidt, "Evolving Feature Model Configurations in Software Product Lines," Journal of Systems and Software, Volume 87, 2014, pp. 119-136.

J109 Akram Hakiri, Aniruddha S. Gokhale, Pascal Berthou, Douglas C. Schmidt, Thierry Gayraud, Software-Defined Networking: Challenges and Research Opportunities for the Future Internet," Journal of Computer Networks, Volume 75, 2014, pp. 453-471.

J108 Hamilton Turner, Brian Dougherty, Jules White, Jonathan Preston, Russell Kegley, Douglas C. Schmidt, and Aniruddha Gokhale, "DRE System Performance Optimization with the SMACK Cache Efficiency Metric," Elsevier Journal of Systems and Software, Volume 98, 2014, pp. 25-43.

J107 Akram Hakiri, Pascal Berthoua, Aniruddha Gokhale, Douglas C. Schmidt, Gayraud Thierry, "Supporting SIP-based Data Distribution Service End-to-End QoS in WANs," the Elsevier Journal of Systems and Software, Volume 95, September 2014, pp. 100-121.

J106 Jules White, Douglas C. Schmidt, and Mani Golparvar-Fard, "Applications of Augmented Reality," IEEE Proceedings Special issue on Applications of Augmented Reality, Vol 102, No. 2., February 2014, pp. 120-123.

J105 Nickolas H. Guertin, Paul Bruhns, Douglas C. Schmidt, and Adam Porter, "Experiences Using Online War Games to Improve the Business of Naval Systems Acquisition," Cutter Journal of Information Technology Management, Vol. 27, No. 5, May 2014, pp 13-18.

J104 Michael McLendon, Bill Scherlis, and Douglas C. Schmidt, "Addressing Software Sustainment Challenges for the DoD," STSC CrossTalk, The Journal of Defense Software Engineering special issue on Legacy Systems Software, January, volume 27, number 1, 2014, pp. 27-32.

J103 Akram Hakiri, Pascal Berthoua, Aniruddha Gokhale, Douglas C. Schmidt, Gayraud Thierry, "Supporting End-to-end Scalability and Real-time Event Dissemination in the OMG Data Distribution Service over Wide Area Networks," Elsevier Journal of Systems and Software, volume 86, number 10, October, 2013, pp. 2574-2593.

J102 William Otte, Aniruddha Gokhale, and Douglas C. Schmidt, "Efficient and Deterministic Application Deployment in Component-based, Enterprise Distributed, Real-time, and Embedded Systems," Elsevier Journal of Information and Software Technology, Vol. 55, No. 2, Feb 2013, 475-488.

J101 Dr. Douglas Schmidt, Anita Carleton, Erin Harper, Mary Ann Lapham, Ipek Ozkaya, and Linda Parker Gates, 'What Will It Take to Achieve Agility-at-Scale?", Cutter IT Journal, edited by Hillel Glazer, November 2012, pp. 34-39.

J100 Brian Dougherty, Jules White, and Douglas C. Schmidt, "Model-driven Auto-scaling of Green Cloud Computing Infrastructure," the Elsevier International Journal of Future Generation Computing Systems, Special Issue on Green Computing Systems, Volume 28, Number 2, February, 2012 Pages 371-378.

J99 Joe Hoffert, Douglas C. Schmidt, and Aniruddha Gokhale, "Evaluating Timeliness and Accuracy Trade-offs of Supervised Machine Learning for Adapting Enterprise DRE Systems in Dynamic Environments," the International Journal of Computational Intelligence Systems, Volume 4, Number 5, September-October 2011, pp. 806-816.

J98 James Hill, Pooja Varshneya, and Douglas C. Schmidt, "Evaluating Distributed Real-time and Embedded System Test Correctness using System Execution Traces," Central European Journal of Computer Science, Volume 1, Number 2, August 2011, pp. 167-184.

J97 Brian Dougherty, Jules White, and Douglas C. Schmidt, "Automated Software and Hardware Evolution Analysis for Distributed Real-time and Embedded Systems," the Central European Journal of Computer Science, Volume 1, Number 1, July 2011, pp. 36-57.

J96 James Hill, Hunt Sutherland, Paul Stodinger, Thomas Silveria, Douglas C. Schmidt, John Slaby, and Nikita Visnevski, "OASIS: An Architecture for Dynamic Instrumentation of Enterprise Distributed Real-time and Embedded Systems," the International Journal of Computer Systems Science and Engineering, Special Issue on Real-time Systems, Volume 26, Number 6, November 2011, pp. 413-430.

J95 Jules White, Brian Dougherty, Chris Thompson, Douglas C. Schmidt, "ScatterD: Spatial Deployment Optimization with Hybrid Heuristic/Evolutionary Algorithms," ACM Transactions on Autonomous and Adaptive Systems Special Issue on Spatial Computing, Volume 6 Issue 3, September 2011, 18:1-8:25.

J94 Jules White, Chris Thompson, Hamilton Turner, Brian Dougherty, and Douglas C. Schmidt, WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones, Journal of Mobile Networks and Applications, Volume 16 Issue 3, July 2011, Pages 285-303.

J93 Jules White, Brian Dougherty, Richard Schantz, Douglas C. Schmidt, Adam Porter, and Angelo Corsaro, "R&D Challenges and Solutions for Highly Complex Distributed Systems: a Middleware Perspective," the Springer Journal of Internet Services and Applications special issue on the Future of Middleware, Volume 2, Number 3, December 2011, pp. 1-8.

J92 Joe Hoffert, Aniruddha Gokhale, and Douglas C. Schmidt, "Autonomic Adaptation of Publish/Subscribe Middleware in Dynamic Environments," the International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS), 2(4), 1-24, October-December 2011, pp. 1-24.

J91 Joe Loyall, Matt Gillen, Aaron Paulos, Larry Bunch, Marco Carvalho, James Edmondson, Douglas C. Schmidt, Andrew Martignoni, and Asher Sinclair, "Dynamic Policy-Driven Quality of Service in Service-Oriented Information Management Systems," Wiley journal on Software: Practice and Experience, December 2011, volume 41, number 12, pp. 1459-1489.

J90 Michael Stal, Douglas C. Schmidt, and Will Otte, "Efficiently and Transparently Automating Scalable On-demand Activation and Deactivation of Services with the Activator Pattern," Software: Practice and Experience, special issue on Pattern Languages: Addressing Challenges, Edited by Mohamed Fayad and Shivanshu Singh, volume 41, number 10, October 2011, Wiley and Sons, pp. 1-16.

J89 Brian Dougherty, Jules White, Douglas C. Schmidt, Russell Kegley, and Jonathan Preston, "Deployment Optimization for Embedded Flight Avionics Systems," STSC CrossTalk, The Journal of Defense Software Engineering, November/December, volume 24, number 6, 2011, pp. 1-8.

J88 Brian Dougherty, Daniel Guymon, Douglas C. Schmidt, and Jules White, "Overcoming Cellular Connectivity Limitations with M2Blue Autonomic Distributed Data Caching," Autonomic Computing for Computer Society of India Magazine, CSI Communications, August 2011, pp. 12-15.

J87 Friedhelm Wolf, Jaiganesh Balasubramanian, Sumant Tambe, Aniruddha Gokhale, and Douglas C. Schmidt, Supporting Component-based Failover Units in Middleware for Distributed Real-time and Embedded Systems, the Elsevier Journal of System Architectures (JSA): Embedded Systems Design, Special Issue on Real-time and Embedded Systems, May, 2011 pp. 597-613.

J86 Jules White, David Benavides, Douglas C. Schmidt, Pablo Trinidad, Antonio Ruiz-Cortes, Brian Doughtery, "Automated Diagnosis of Feature Model Configurations," The Journal of Systems and Software, Special Issue on Software Product-lines, Volume 83, Issue 7, July, 2010, pp. 1094-1107.

J85 Jules White, Brian Doughtery, and Douglas C. Schmidt, "ASCENT: An Algorithmic Technique for Designing Hardware and Software in Tandem, IEEE Transactions on Software Engineering Special Issue on Search-based Software Engineering, November/December 2010 (vol. 36 no. 6), pp. 838-851.

J84 Joe Hoffert, Daniel Mack, and Douglas Schmidt, "Integrating Machine Learning Techniques to Adapt Protocols for QoS-enabled Distributed Real-time and Embedded Publish/Subscribe Middleware," International Journal of Network Protocols and Algorithms (NPA): Special Issue on Data Dissemination for Large-scale Complex Critical Infrastructures, Volume 2, Number 3, 2010, pp. 37-69.

J83 James Hill, James Edmondson, Aniruddha Gokhale, and Douglas C. Schmidt, "Tools for Continuously Evaluating Distributed System Qualities," IEEE Software, July/August, 2010, Volume 27, Number 4, pp. 65-71.

J82 James Edmondson and Douglas C. Schmidt, Multi-Agent Distributed Adaptive Resource Allocation (MADARA), International Journal of Communication Networks and Distributed Systems (IJCNDS), Special Issue on: Grid Computing, Edited by Michal Wozniak and Krzysztof Walkowiak, Volume 5, Number 3, 2010, pp. 229-245.

J81 Jules White, Christin Groba, Sibohan Clarke, Brian Dougherty, Chris Thompson, and Douglas C. Schmidt, "R&D Challenges and Solutions for Mobile Cyber-Physical Applications and Supporting Internet Services," the Springer Journal of Internet Services and Applications, Volume 1, Number 1, 2010, pp. 45-56.

J80 Jules White, Jeff Gray, and Douglas C. Schmidt, "Constraint-based Model Weaving," Transactions on Aspect-Oriented Software Development, Special Issue on Aspects and Model Driven Engineering, eds. Robert France and Jean-Marc Jezequel, pp. 153-190, Volume 5560, Number 6, 2009.

J79 Jules White, Harrison Strowd, and Douglas C. Schmidt, "Creating Self-healing Service Compositions with Feature Modeling and Microrebooting," the International Journal of Business Process Integration and Management (IJBPIM), Special issue on Model-Driven Service-Oriented Architectures, Inderscience Publishers, pp. 35-46, Volume 4, Number 1, 2009.

J78 Nishanth Shankaran, John Kinnebrew, Xenofon Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Gautam Biswas, "An Integrated Planning and Adaptive Resource Management Architecture for Distributed Real-time Embedded Systems," IEEE Transactions on Computers, Special Issue on Autonomic Network Computing, Special Issue on Autonomic Network Computing, volume 58, number 11, pp. 1485-1498, November 2009.

J77 Jules White, Brian Doughtery, and Douglas C. Schmidt, "Selecting Highly Optimal Architectural Feature Sets with Filtered Cartesian Flattening," the Journal of Software and Systems, Special Issue on Design Decisions and Design Rationale in Software Architecture, Volume 82, Issue 8, pp. 1268-1284, August 2009.

J76 Jules White, James, Hill, Sumant Tambe, Jeff Gray, Aniruddha Gokhale, and Douglas C. Schmidt "Improving Domain-specific Language Reuse through Software Product-line Configuration Techniques, IEEE Software Special Issue: Domain-Specific Languages and Modeling, vol. 26, no. 4, pp. 47-53, July/August 2009.

J75 Jules White and Douglas C. Schmidt, "Automating Deployment Planning with an Aspect Weaver," IET Software Journal Special Issue on Domain-specific Aspect Languages, Volume 3, Issue 3 , p. 167-183, June 2009.

J74 Shanshan Jiang, Yuan Xue, and Douglas C. Schmidt, "Minimum Disruption Service Composition and Recovery in Mobile Ad Hoc Networks, Elsevier Computer Networks Journal, Special Issue on Autonomic and Self-Organizing Systems, Volume 53, Issue 10, Pages 1649-1665, 2009.

J73 Nishanth Shankaran, Douglas C. Schmidt, Xenofon D. Koutsoukos, Yingming Chen, and Chenyang Lu, "Design and Performance Evaluation of an Adaptive Resource Management Framework for Distributed Real-time and Embedded Systems," EURASIP Journal on Embedded Systems (EURASIP JES): Special issue on Operating System Support for Embedded Real-Time Applications, Edited by Alfons Crespo, Ismael Ripoll, Michael Gonzalez Harbour, and Giuseppe Lipari, 2008, Pgs. 47-66.

J72 Aniruddha Gokhale, Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Arvind Krishna, and George T. Edwards, Gan Deng, Emre Turkay, Jeffrey Parsons, and Douglas C. Schmidt, Model Driven Middleware: A New Paradigm for Deploying and Provisioning

Distributed Real-time and Embedded Applications, Elsevier Journal of Science of Computer Programming: Special Issue on Foundations and Applications of Model Driven Architecture (MDA), Edited by Mehmet Aksit, Volume 73, Issue 1, 1 September 2008, Pgs. 39-58.

J71 Nishanth Shankaran, Xenofon Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Yuan Xue, "Hierarchical Control of Multiple Resources in Distributed Real-time and Embedded Systems," the Springer Real-time Systems Journal, Volume 39, Numbers 1-3, August, 2008, pgs. 237-282.

J70 Douglas C. Schmidt, Angelo Corsaro, and Hans Van'T Hag, "Addressing the Challenges of Tactical Information Management in Net-Centric Systems with DDS," CrossTalk special issue on Distributed Software Development, pgs. 24-29, May 2008.

J69 Jules White, Douglas C. Schmidt, Egon Wuchner, and Andrey Nechypurenko, "Automatically Composing Reusable Software Components for Mobile Devices," Journal of the Brazilian Computer Society (JBCS), Special Issue in Software Reuse: Methods, Processes, Tools and Experiences, Sociedade Brasileira de Computacao, Porto Alegre, Volume 14, Number 1, pgs. 25-44, March, 2008.

J68 Jules White, Douglas C. Schmidt, Andrey Nechypurenko, and Egon Wuchner, "Model Intelligence: an Approach to Modeling Guidance," UPGRADE Journal, Volume 9, Number 2, pgs. 22-28, April 2008.

J67 Douglas C. Schmidt, "Beyond Objects: Evaluating Technologies for Developing Distributed Systems," Computer Society of India Communications, Special Issue on OO Technologies, edited by Debasish Jana, pgs. 30-37, February 2008.

J66 Jules White, Douglas C. Schmidt, and Aniruddha Gokhale, "Simplifying Autonomic Enterprise Java Bean Applications via Model-driven Engineering and Simulation," The Journal of Software and System Modeling, Volume 7, Number 1, February, 2008, pgs. 3-23.

J65 Cemal Yilmaz, Adam Porter, Arvind S. Krishna, Atif Memon, Douglas C. Schmidt, and Aniruddha Gokhale, "Reliable Effects Screening: A Distributed Continuous Quality Assurance Process for Monitoring Performance Degradation in Evolving Software Systems," IEEE Transactions on Software Engineering Vol. 33, No. 8, August 2007, pgs. 510-525.

J64 Frank Buschmann, Kevlin Henning, and Douglas C. Schmidt, "Past, Present, and Future Trends in Software Patterns," IEEE Software special issue on Patterns, Vol. 24, No. 4, July/August, 2007, pgs. 31-37.

J63 Patrick Lardieri, Jaiganesh Balasubramanian, Douglas C. Schmidt, Gautam Thaker, Aniruddha Gokhale, and Tom Damiano, A Multi-layered Resource Management Framework for Dynamic Resource Management in Enterprise DRE Systems, the Journal of Systems and Software: special issue on Dynamic Resource Management in Distributed Real-Time Systems, editors C. Cavanaugh and F. Drews and L. Welch, Vol 80, Issue 7, July 2007, pgs. 984-996.

J62 Janos Sztipanovits, John Bay, Larry Rohrbough, Shankar Sastry, Douglas C. Schmidt, Don Wilson, and Don Winters, "Escher: A New Technology Transitioning Model," IEEE Computer, Vol. 40, No. 3, March 2007, pgs. 90-92.

J61 Venkita Subramonian, Gan Deng, Christopher Gill, Jaiganesh Balasubramanian, Liang-Jui Shen, William Otte, Douglas C. Schmidt, Aniruddha Gokhale, and Nanbor Wang, "The Design and Performance of Component Middleware for QoS-enabled Deployment and Configuration of DRE Systems," Elsevier Journal of Systems and Software, Special Issue Component-Based Software Engineering of Trustworthy Embedded Systems, pp. 668-677, volume 80, number 5, March, 2007.

J60 Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Jeff Parsons, Aniruddha Gokhale, and Douglas C. Schmidt, "A Platform-Independent Component Modeling Language for Distributed Real-time and Embedded Systems," Elsevier Journal of Computer and System Sciences, Volume 73, Number 2, March 2007, pgs. 171 - 185.

J59 Adam Porter, Atif Memon, Cemal Yilmaz, Douglas C. Schmidt, and Bala Natarajan, "Skoll: A Process and Infrastructure for Distributed Continuous Quality Assurance, IEEE Transactions on Software Engineering, 2007, Vol. 33, No. 2, February 2007, pgs. 124-141.

J58 Richard E. Schantz, Douglas C. Schmidt, Joseph P. Loyall, and Craig Rodrigues, "Controlling Quality-of-Service in Distributed Real-time and Embedded Systems via Adaptive Middleware," the Wiley Software Practice and Experience journal special issue on Experiences with

Auto-adaptive and Reconfigurable Systems, co-editors Mehmet Aksit, Zied Choukair, and Tzilla Elrad, vol. 36, no. 11-12, September 2006, pgs. 1189 - 1208.

J57 Douglas C. Schmidt, "Model-Driven Engineering, IEEE Computer, Vol. 39. No. 2, February 2006, pp. 41-47.

J56 Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, and Venkatesh Prasad Ranganat, "Towards Highly Optimized Real-time Middleware for Software Product-line Architectures," ACM SIGBED Review, Volume 3, No. 1, January 2006, pgs. 12-16.

J55 Gabor Madl, Sherif Abdelwahed, and Douglas C. Schmidt, "Verifying Distributed Real-time Properties of Embedded Systems via Graph Transformations and Model Checking, Real-time Systems Journal, vol 33, no. 1-3, pgs. 77-100, July 2006.

J54 Cemal Yilmaz, Adam Porter, Atif Memon, Arvind S. Krishna, Douglas C. Schmidt, and Aniruddha Gokhale, Techniques and Processes for Improving the Quality and Performance of Open-Source Software, Software Process - Improvement and Practice Journal: Special Issue on Free/Open Source Software Processes, vol 11, no 2, May 2006, pgs. 163-176.

J53 Krishnakumar Balasubramanian, Arvind S. Krishna, Emre Turkay, Jaiganesh Balasubramanian, Jeff Parsons, Aniruddha Gokhale, and Douglas C. Schmidt, "Applying Model-Driven Development to Distributed Real-time and Embedded Avionics Systems, the International Journal of Embedded Systems, special issue on Design and Verification of Real-Time Embedded Software, April 2005.

J52 Arvind S. Krishna, Cemal Yilmaz, Adam Porter, Atif Memon, Douglas C. Schmidt, and Aniruddha Gokhale, "Distributed Continuous Quality Assurance Process for Evaluating QoS of Performance Intensive Software," Studia Informatica Universalis, Volume 4, No. 1, pp. 53-72, March 2005.

J51 Janos Sztipanovits, Gautam Biswas, Ken Frampton, Andy Gokhale, Larry Howard, Gabor Karsai, John Koo, Xenofon Koutsoukos, and Douglas C. Schmidt, "Introducing Embedded Software and Systems Education and Advanced Learning Technology in an Engineering Curriculum," ACM Transactions in Embedded Computing Systems - Special Issue on Education, edited by Alan Burns and Alberto Sangiovanni-Vincentelli, Vol 4, No. 1, pp. 549-568, August 2005.

J50 Arvind S. Krishna, Nanbor Wang, Balachandran Natarajan, Aniruddha Gokhale, Douglas C. Schmidt and Gautam Thaker, "CCMPerf: A Benchmarking Tool for CORBA Component Model Implementations", The International Journal of Time-Critical Computing Systems, Springer, Vol. 29, No. 2-3, pp. 281-308, March-April 2005.

J49 Chris Gill, Jeanna M. Gossett, David Corman, Joseph P. Loyall, Richard E. Schantz, Michael Atighetchi, and Douglas C. Schmidt, "Integrated Adaptive QoS Management in Middleware: An Empirical Case Study," The International Journal of Time-Critical Computing Systems, Springer, Vol. 29, Nos. 2-3, pp. 101-130, March-April 2005.

J48 Aniruddha Gokhale, Balachandran Natarajan, Douglas C. Schmidt, and Joseph Cross, "Towards Real-time Fault-Tolerant CORBA Middleware," Cluster Computing: the Journal on Networks, Software, and Applications Special Issue on Dependable Distributed Systems, edited by Alan George, Volume 7, Number 4, October 2004.

J47 Arvind S. Krishna, Cemal Yilmaz, Atif Memon, Adam Porter, Douglas C. Schmidt, Aniruddha Gokhale, and Balachandran Natarajan, "Preserving Distributed Systems Critical Properties: a Model-Driven Approach," the IEEE Software special issue on the Persistent Software Attributes, Nov/Dec 2004.

J46 Christopher Gill, Douglas C. Schmidt, Yamuna Krishnamurthy, Irfan Pyarali, Louis Mgeta, Yuanfang Zhang, and Stephen Torri, "Enhancing the Adaptivity of Distributed Real-time and Embedded Systems via Standard QoS-enabled Dynamic Scheduling Middleware," the Journal of the Brazilian Computer Society (JCBS) special issue on Adaptive Software Systems, Volume 10, Number 1, pp. 19-30, 2004.

J45 Douglas C. Schmidt, Aniruddha Gokhale, and Balachandran Natarajan, "Frameworks: Why They Are Important and How to Apply Them Effectively," ACM Queue magazine, Volume 2, Number 5, July/August 2004.

J44 Douglas C. Schmidt, Richard Schantz, Aniruddha Gokhale, and Joe Loyall, "Middleware R&D Challenges for Distributed Real-time and Embedded Systems," ACM SIGBED Review, Volume 1, No. 1, April 2004.

J43 Angelo Corsaro and Douglas C. Schmidt, "The Design and Performance of Real-time Java Middleware," Special Issue on Middleware for the IEEE Transactions on Parallel and Distributed Systems, guest editor Rachid Guerraoui, Volume 14, Number 11, November 2003.

J42 Irfan Pyarali, Douglas C. Schmidt, and Ron Cytron, "Techniques for Enhancing Real-time CORBA Quality of Service," the IEEE Proceedings Special Issue on Real-time Systems, co-editors Yann-Hang Lee and C. M. Krishna, Volume 91, Number 7, July 2003.

J41 Nanbor Wang, Douglas C. Schmidt, Aniruddha Gokhale, Christopher D. Gill, Balachandran Natarajan, Craig Rodrigues, Joseph P. Loyall, and Richard E. Schantz, "Total Quality of Service Provisioning in Middleware and Applications," Elsevier Journal of Microprocessors and Microsystems, Volume 26, Number 9-10, January 2003.

J40 Christopher D. Gill, Douglas C. Schmidt, and Ron Cytron, "Multi-Paradigm Scheduling for Distributed Real-Time Embedded Computing," IEEE Proceedings Special Issue on Modeling and Design of Embedded Systems, Volume 91, Number 1, January, 2003.

J39 Aniruddha Gokhale, Bala Natarajan, Douglas C. Schmidt, and Nanbor Wang, "Modeling and Synthesis of Middleware Components," Communications of the ACM, special issue on Enterprise Components, Services and Business Rules, edited by Ali Arsanjani, October 2002.

J38 Douglas C. Schmidt and Carlos O'Ryan, "Patterns and Performance of Distributed Real-time and Embedded Publisher/Subscriber Architectures," the Journal of Systems and Software, Special Issue on Software Architecture – Engineering Quality Attributes, edited by Jan Bosch and Lars Lundberg, October 2002.

J37 Douglas C. Schmidt, "R&D Advances in Middleware for Distributed, Real-time and Embedded Systems," Communications of the ACM, Volume 45, Number 6, June 2002, edited by Gul Agha.

J36 Carlos O'Ryan, Douglas C. Schmidt, and J. Russell Noseworthy, "Patterns and Performance of a CORBA Event Service for Large-scale Distributed Interactive Simulations," International Journal of Computer Systems Science and Engineering, CRL Publishing, Volume 17, Number 2, March, 2002.

J35 Douglas C. Schmidt, Bala Natarajan, Aniruddha Gokhale, Nanbor Wang, and Chris Gill, "TAO: A Pattern-Oriented Object Request Broker for Distributed Real-time and Embedded Systems," IEEE Distributed Systems Online, Volume 3, Number 2, February, 2002.

J34 Douglas C. Schmidt, Rick Schantz, Mike Masters, Joseph Cross, David Sharp, and Lou DiPalma, "Towards Adaptive and Reflective Middleware for Network-Centric Combat Systems," CrossTalk, November, 2001.

J33 Nanbor Wang, Douglas C. Schmidt, Ossama Othman, and Kirthika Parameswaran, "Evaluating Meta-Programming Mechanisms for ORB Middleware," *IEEE Communications Magazine* special issue on "Evolving Communications Software: Techniques and Technologies," co-edited by Bill Opdyke and Algirdas Pakstas, Volume 39, Number 10, October, 2001.

J32 Nanbor Wang, Douglas C. Schmidt, Kirthika Parameswaran, and Michael Kircher, "Towards a Reflective Middleware Framework for QoS-enabled CORBA Component Model Applications," *IEEE Distributed Systems Online* special issue on Reflective Middleware, July, 2001.

J31 Chris Gill, David Levine, and Douglas C. Schmidt, "The Design and Performance of a Real-Time CORBA Scheduling Service," The *Real-time Systems, The International Journal of Time-Critical Computing Systems*, special issue on Real-Time Middleware, Kluwer Academic Publishers, guest editor Wei Zhao, Volume 20, Number 2, March 2001.

J30 Douglas C. Schmidt, Sumedh Mungee, Sergio Flores-Gaitan, and Aniruddha Gokhale, "Software Architectures for Reducing Priority Inversion and Non-determinism in Real-time Object Request Brokers," *Journal of Real-time Systems*, Kluwer, Vol. 21, No. 2, 2001.

J29 Ossama Othman, Carlos O'Ryan, and Douglas C. Schmidt, "An Efficient Adaptive Load Balancing Service for CORBA," *IEEE Distributed Systems Online*, March, 2001.

J28 Ossama Othman, Carlos O'Ryan, and Douglas C. Schmidt "The Design of an Adaptive CORBA Load Balancing Service, *IEEE Distributed Systems Online*, April, 2001.

J27 Carlos O'Ryan, Douglas C. Schmidt, Fred Kuhns, Marina Spivak, Jeff Parsons Irfan Pyarali, and David L. Levine, "Evaluating Policies and Mechanisms to Support Distributed Real-Time Applications with CORBA," *Concurrency and Computation: Practice and Experience* (Special Issue on Distributed Objects and Applications), Wiley and Sons, Vol. 13, No. 2, February, 2001.

J26 Douglas C. Schmidt, Vishal Kachroo, Yamuna Krisnamurthy, and Fred Kuhns, "Developing Next-generation Distributed Applications with QoS-enabled DPE Middleware," *IEEE Communications magazine*, edited by Abdi Modarressi and Sheshadri Mohan, Vol 17, No. 10, October, 2000.

J25 Douglas C. Schmidt and Fred Kuhns, "An Overview of the Real-time CORBA Specification," *IEEE Computer*, Special Issue on Object-Oriented Real-time Distributed Computing, edited by Eltefaat Shokri and Philip Sheu, June 2000.

J24 James Hu and Douglas C. Schmidt, Developing Flexible and High-performance Web Servers with Frameworks and Patterns, Symposium on Frameworks, *ACM Computing Surveys*, (Fayad and Wegner, eds.) Vol. 32(1es), March 2000.

J23 Fred Kuhns, Douglas C. Schmidt, Carlos O'Ryan, and David L. Levine, "Supporting High-performance I/O in QoS-enabled ORB Middleware," *Cluster Computing: the Journal on Networks, Software, and Applications*, Volume 3, Number 3, 2000.

J22 Irfan Pyarali, Carlos O'Ryan, Douglas C. Schmidt, Nanbor Wang, Vishal Kachroo, and Aniruddha Gokhale, "Using Principle Patterns to Optimize Real-time ORBs," *IEEE Concurrency*, Object-Oriented Systems track, edited by Murthy Devarakonda, Volumn 8, Number 1, January-March 2000.

J21 James Hu, Irfan Pyarali, and Douglas C. Schmidt, "The Object-Oriented Design and Performance of JAWS: A High-performance Web Server Optimized for High-speed Networks," The *Parallel and Distributed Computing Practices* journal, special issue on Distributed Object-Oriented Systems, edited by Maria Cobb, Vol. 3, No. 1, March 2000.

J20 Andy Gokhale and Douglas C. Schmidt, "Optimizing a CORBA IIOP Protocol Engine for Minimal Footprint Multimedia Systems," *IEEE Journal on Selected Areas in Communications* special issue on Service Enabling Platforms for Networked Multimedia Systems, September, 1999.

J19 Douglas C. Schmidt and Chris Cleeland, "Applying Patterns to Develop Extensible and Maintainable ORB Middleware," *IEEE Communications Magazine*, April, 1999.

J18 Irfan Pyarali and Douglas C. Schmidt, "An Overview of the CORBA Portable Object Adapter," Special Issue on CORBA in the ACM *StandardView* magazine, March, 1999.

J17 Prashant Jain, Seth Widoff, and Douglas C. Schmidt, "The Design and Performance of Med-Java, A Distributed Electronic Medical Imaging System Developed with Java Applets and Web Tools," IEE/BCS *Distributed Systems Engineering Journal*, Volume 5, No. 4, December 1998.

J16 Douglas C. Schmidt, "Evaluating Architectures for Multi-threaded CORBA Object Request Brokers," *Communications of the ACM*, Special Issue on CORBA, ACM, edited by Krishnan Seetharaman, Volume 41, No. 10, October 1998.

J15 Andy Gokhale and Douglas C. Schmidt, "Measuring and Optimizing CORBA Latency and Scalability Over High-speed Networks," *IEEE Transactions on Computing*, April, 1998.

J14 Douglas C. Schmidt and James Hu, "Developing Flexible and High-performance Web Servers with Frameworks and Patterns," *Computing Surveys*, ACM, Vol. 29, March 1998.

J13 Douglas C. Schmidt, David Levine, and Sumedh Mungee, "The Design of the TAO Real-Time Object Request Broker," *Computer Communications*, Special Issue on Building Quality of Service into Distributed System, Elsevier Science, April, 1998.

J12 Guru Parulkar, Douglas C. Schmidt, Eileen Kraemer, Jon Turner, Anshul Kantawala, "An Architecture for Monitoring, Visualization, and Control and Gigabit Networks," *IEEE Network*, September/October, 1997.

J11 Douglas C. Schmidt, "Lessons Learned Building Reusable OO Frameworks for Distributed Software," *Communications of the ACM* Special Issue on OO Application Frameworks, ACM, Vol. 40, No. 10, October, 1997.

J10 Douglas C. Schmidt, "Applying Patterns to Meet the Challenges of Concurrent Software," *IEEE Concurrency*, Special Edition on Software Engineering for Parallel and Distributed Systems, Vol. 5, No. 3, August, 1997.

J9 Douglas C. Schmidt, Andy Gokhale, Tim Harrison, and Guru Parulkar, "A High-performance Endsystem Architecture for Real-time CORBA," *IEEE Communications Magazine*, Vol. 14, No. 2, February, 1997.

J8 Silvano Maffeis and Douglas C. Schmidt, "Constructing Reliable Distributed Communication Systems with CORBA," *IEEE Communications Magazine*, Vol. 14, No. 2, February, 1997.

J7 Douglas C. Schmidt, "Using Design Patterns to Develop Reuseable Object-Oriented Software," *ACM Computing Surveys*, Vol. 28, No. 4es, December 1996.

J6 Irfan Pyarali, Douglas C. Schmidt, and Tim Harrison, "Design and Performance of an Object-Oriented Framework for High-Speed Electronic Medical Imaging," *USENIX Computing Systems*, November/December, Vol. 9, No. 4, 1996.

J5 Douglas C. Schmidt, "A Family of Design Patterns for Application-level Gateways," *The Journal of Theory and Practice of Object Systems* (Special Issue on Patterns and Pattern Languages), Wiley and Sons, Vol. 2, No. 1, 1996.

J4 Douglas C. Schmidt, "Experience Using Design Patterns to Develop Reuseable Object-Oriented Communication Software," *Communications of the ACM Special Issue on Object-Oriented Experiences*, ACM, Vol. 38, No. 10, October, 1995, pp 65-74.

J3 Douglas C. Schmidt and Tatsuya Suda, "An Object-Oriented Framework for Dynamically Configuring Extensible Distributed Systems," *Distributed Systems Engineering Journal* (Special issue on Configurable Distributed Systems), IEE, Vol. 2, December, 1994, pp. 280-293.

J2 Douglas C. Schmidt, Donald F. Box, and Tatsuya Suda, "ADAPTIVE: A Dynamically Assembled Protocol Transformation, Integration, and eValuation Environment," *Journal of Concurrency: Practice and Experience*, Wiley and Sons, Ltd., Vol. 5, No. 4, June, 1993, pp. 269-286.

J1 Douglas C. Schmidt and Tatsuya Suda, "Transport System Architecture Services for High-Performance Communication Systems," *Journal of Selected Areas of Communications special-issue on Protocols for Gigabit Networks*, IEEE, Vol. 11, No. 4, May, 1993, pp. 489-506.

- **Book Publications and Book Chapters**

  – **Books Authored**

BA5 Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt, *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, Wiley and Sons, 2007.

BA4 Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt, *Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing*, Wiley and Sons, 2007.

BA3 Steve Huston and Douglas C. Schmidt, *C++ Network Programming: Systematic Reuse with ACE and Frameworks*, Addison-Wesley Longman, 2003.

BA2 Douglas C. Schmidt and Steve Huston, *C++ Network Programming: Mastering Complexity with ACE and Patterns*, Addison-Wesley Longman, 2002.

BA1 Douglas C. Schmidt, Michael Stal, Hans Rohert, and Frank Buschmann, *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*, John Wiley and Sons, 2000.

  – **Book-length Reports Authored**

BR4 Werner Dahm, Douglas Schmidt, et al. "Cyber Situational Awareness," SAB-TR-12-01, 1 October 2012.

BR3 Alan Eckbreth, Charles Saff, Kevin Connolly, Natalie Crawford, Chris Eick, Mark Goorsky, Neil Kacena, David Miller, Robert Schafrik, Douglas Schmidt, "Sustaining Air Force Aging Aircraft into the 21st Century," SAB-TR-11-01 1 August 2011.

BR2 William Scherlis, Robert Behler, Barry Boehm, Lori Clarke, Michael Cusumano, Mary Ann Davidson, Larry Druffel, Russell Frew, James Larus, Greg Morrisett, Walker Royce, Douglas C. Schmidt, John Stenbit, Kevin Sullivan, *Critical Code Software Producibility for Defense*, Committee for Advancing Software-Intensive Systems Producibility, National Research Council of the National Academies, Washington, D.C.

BR1 Linda Northrop, Peter Feiler, Richard P. Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein, Linda Northrop, Douglas Schmidt, Kevin Sullivan, and Kurt Wallnau *Ultra-Large-Scale Systems: Software Challenge of the Future*, Software Engineering Institute, June 2006.

  – **Books Edited**

BE3 Co-editor of *Building Application Frameworks: Object-Oriented Foundations of Framework Design*, John Wiley & Sons, 1999 (co-editors are Mohamed Fayad and Ralph Johnson), ISBN 0-471-24875-4

BE2 Co-editor of *Implementing Application Frameworks: Object-Oriented Frameworks at Work*, John Wiley & Sons, 1999 (co-editors are Mohamed Fayad and Ralph Johnson), ISBN 0-471-25201-8

BE1 Co-editor of *Pattern Languages of Program Design*, Addison-Wesley, 1995 (co-editor is Jim Coplien, Bell Labs).

– **Book Chapters**

BC64 Michael Walker, Douglas C. Schmidt, and Abhishek Dubey, "Testing at Scale of IoT Blockchain Applications," *Role of Blockchain Technology in IoT Applications*, Volume 115, 1st Edition, edited by Shiho Kim, Ganesh Chandra Deka, and Peng Zhang, 2019.

BC63 Peng Zhang, Douglas C. Schmidt, Jules White, and Abhishek Dubey, "Consensus Mechanisms and Information Security Technologies," *Role of Blockchain Technology in IoT Applications*," Volume 115, 1st Edition, edited by Shiho Kim, Ganesh Chandra Deka, and Peng Zhang, 2019.

BC62 Peng Zhang, Douglas C. Schmidt, Jules White, and Gunther Lenz, "Blockchain Technology Use Cases in Healthcare," *Blockchain Technology: Platforms, Tools, and Use Cases*, edited by Ganesh Deka, 2018.

BC61 Michael Walker, Douglas C. Schmidt, and Jules White, "An Elastic Platform for Large-scale Assessment of Software Assignments for MOOCs (EPLASAM)," User-Centered Design Strategies for Massive Open Online Courses (MOOCs), edited by Ricardo Mendoza-Gonzalez, IGI Global, 2016.

BC60 Joseph Hoffert, Douglas C. Schmidt, and Aniruddha Gokhale, "Quantitative Productivity Analysis of a Domain-Specific Modeling Language," in the *Handbook of Research on Innovations in Systems and Software Engineering*, IGI Global, Aug 2014, pp. 313-344.

BC59 Will Otte, Douglas C. Schmidt, and Aniruddha Gokhale, "Performance and Scalability of a Large-scale Deployment and Configuration Framework," *The Performance of Open Source Applications*, edited by Tony Arkles and Tavish Armstrong, O'Reilly, 2013.

BC58 Chris Thompson, Jules White, and Douglas C. Schmidt, "Analyzing Mobile Application Software Power Consumption via Model-Driven Engineering," *Advances and Applications in Model-Driven Software Engineering*, edited by Vicente Garcia Diaz, IGI Global, 2013.

BC57 James Edmondson and Douglas C. Schmidt, "Towards Accurate Simulation of Large-Scale Systems via Time Dilation," *Real-time Simulation Technologies: Principles, Methodologies, and Applications*, edited by Katalin Popovici and Pieter J. Mosterman, CRC Press, 2012.

BC56 James Hill and Douglas C. Schmidt, "Using Test Clouds to Enable Continuous Integration Testing of Distributed Real-time and Embedded System Applications," *Software Testing in the Cloud: Perspectives on an Emerging Discipline.* Edited by Scott Tilley and Tauhida Parveen, IGI Global, 2012.

BC55 Angelo Corsaro and Douglas C. Schmidt, "The Data Distribution Service: The Communication Middleware Fabric for Scalable and Extensible Systems-of-Systems," *System of Systems*, edited by Adrian V. Gheorghe, InTech, 2012.

BC54 Joe Hoffert and Douglas C. Schmidt, "Improving Software Development Productivity for QoS Policy Configurations," *Model-Driven Domain Analysis and Software Development: Architectures and Functions*, a book edited by Janis Osis and Erika Asnina, 2011.

BC53 Nilabja Roy and Douglas C. Schmidt, "Model-Driven Performance Evaluation of Web Application Portals," *Model-Driven Domain Analysis and Software Development: Architectures and Functions*, a book edited by Janis Osis and Erika Asnina, 2011.

BC52 Brian Dougherty, Jules White, and Douglas C. Schmidt, "MDA-based Configuration of Distributed Real-time and Embedded Systems," *Model-Driven Domain Analysis and Software Development: Architectures and Functions*, a book edited by Janis Osis and Erika Asnina, 2011.

BC51 Hamilton Turner, Jules White, Brian Dougherty, and Douglas C. Schmidt, "Building Mobile Sensor Networks Using Smartphones and Web Services: Ramifications and Development Challenges," *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*, edited by Maria Manuela Cruz-Cunha and Fernando Moreira, IGI Global, Hershey, PA, USA 2009.

BC50 Gan Deng, Jeff Gray, Douglas C. Schmidt, Yuehua Lin, Aniruddha Gokhale, and Gunther Lenz, "Evolution in Model-Driven Software Product-Line Architectures," *Software Ap-*

*plications: Concepts, Methodologies, Tools, and Applications*, edited by Pierre F. Tiako, 2009.

BC49 Jules White, Douglas C. Schmidt, Andrey Nechypurenko, and Egon Wuchner, "Reducing the Complexity of Modeling Large Software Systems," *Software Applications: Concepts, Methodologies, Tools, and Applications*, edited by Pierre F. Tiako, 2009.

BC48 Jules White, Brian Doughtery, Harrison Strowd, and Douglas C. Schmidt, "Using Filtered Cartesian Flattening and Microrebooting to Build Enterprise Applications with Self-adaptive Healing," *Software Engineering for Self-Adaptive Systems*, edited by Betty H. C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, 2009.

BC47 Jeff Gray, Sandeep Neema, Jing Zhang, Yuehua Lin, Ted Bapty, Aniruddha Gokhale, and Douglas C. Schmidt, "Concern Separation for Adaptive QoS Modeling in Distributed Real-Time Embedded Systems," *Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation*, edited by Joa M. Fernandes and Luis Gomes, 2009.

BC46 Rick Schantz and Douglas C. Schmidt, "Middleware for Distributed Systems," Wiley *Encyclopedia of Computer Science and Engineering*, edited by Benjamin Wah, 2008.

B45 Jules White, Andrey Nechypurenko, Egon Wuchner, and Douglas C. Schmidt, "Reducing the Complexity of Optimizing Large-scale Systems by Integrating Constraint Solvers with Graphical Modeling Tools," *Designing Software-Intensive Systems: Methods and Principles*, Edited by Dr. Pierre F. Tiako, Langston University, OK, April 2008.

B44 Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale, Jeff Gray, Yuehua Lin, and Gunther Lenz, "Evolution in Model-Driven Software Product-line Architecture, *Designing Software-Intensive Systems: Methods and Principles*, Edited by Dr. Pierre F. Tiako, Langston University, OK, April 2008.

B43 Daniel G. Waddington, Nilabja Roy, and Douglas C. Schmidt, "Dynamic Analysis and Profiling of Multi-threaded Systems," *Designing Software-Intensive Systems: Methods and Principles*, Edited by Dr. Pierre F. Tiako, Langston University, OK, April 2008.

B42 Krishnakumar Balasubramanian, Douglas C. Schmidt, Zoltan Molnar, and Akos Ledeczi, "System Integration via Model-Driven Engineering," *Designing Software-Intensive Systems: Methods and Principles* Edited by Dr. Pierre F. Tiako, Langston University, OK, April 2008.

B41 James Hill, Douglas C. Schmidt,, and John Slaby, "System Execution Modeling Tools for Evaluating the Quality of Service of Enterprise Distributed Real-time and Embedded System",*Designing Software-Intensive Systems: Methods and Principles*, Edited by Dr. Pierre F. Tiako, Langston University, OK, April 2008.

B40 Gan Deng, Chris Gill, Douglas C. Schmidt, and Nanbor Wang, "QoS-enabled Component Middleware for Distributed Real-Time and Embedded Systems," Handbook of Real-Time and Embedded Systems (I. Lee, J. Leung, and S. Son, eds.), CRC Press, 2007.

B39 William Otte and Douglas C. Schmidt, "Labor-Saving Architecture: an Object-Oriented Framework for Networked Software," *Beautiful Code*, edited by Greg Wilson, O'Reilly, 2007.

BC38 Irfan Pyarali, Carlos O'Ryan, and Douglas C. Schmidt, "A Pattern Language for Efficient, Predictable, Scalable, and Flexible Dispatching Components," Pattern Language of Program Design 5 book, Addison-Wesley, Reading, MA, 2006.

BC37 Douglas C. Schmidt, Krishnakumar Balasubramanian, Arvind S. Krishna, Emre Turkay, and Aniruddha Gokhale, Model-driven Development of Component-based Distributed Real-time and Embedded Systems, Model Driven Engineering for Distributed Real-time and Embedded Systems, edited by Sebastien Gerard, Joel Champea, and Jean-Philippe Babau, Hermes, 2005.

BC36 Gabriele A. Trombetti, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, Gurdip Singh, and Jesse Greenwald, "An Integrated Model-driven Development Environment for Composing and Validating Distributed Real-time and Embedded Systems," *Model-driven Software Development: Volume II of Research and Practice in Software Engineering*, edited by Sami Beydeda, Matthias Book, and Volker Gruhn, Springer-Verlag, 2005.

BC35 Arvind S. Krishna, Douglas C. Schmidt, Ray Klefstad, and Angelo Corsaro, "Real-time CORBA Middleware," in *Middleware for Communications*, edited by Qusay Mahmoud, Wiley and Sons, New York, 2003.

BC34 Nanbor Wang, Douglas C. Schmidt, Aniruddha Gokhale, Craig Rodrigues, Balachandran

Natarajan, Joseph P. Loyall, Richard E. Schantz, and Christopher D. Gill, "QoS-enabled Middleware," in *Middleware for Communications*, edited by Qusay Mahmoud, Wiley and Sons, New York, 2003.

BC33 Aniruddha Gokhale, Douglas C. Schmidt, Balachandran Natarajan, Jeff Gray, and Nanbor Wang, "Model Driven Middleware," in *Middleware for Communications*, edited by Qusay Mahmoud, Wiley and Sons, New York, 2003.

BC32 Jeff Gray, Janos Sztipanovits, Ted Bapty, Sandeep Neema, Aniruddha Gokhale, and Douglas C. Schmidt, "Two-level Aspect Weaving to Support Evolution of Model-Based Software," in *Aspect-Oriented Software Development,* edited by Robert Filman, Tzilla Elrad, Mehmet Aksit, and Siobhan Clarke, Reading, Massachusetts: Addison-Wesley, 2003.

BC31 Joseph K. Cross and Douglas C. Schmidt "Applying the Quality Connector Pattern to Optimize Distributed Real-time and Embedded Middleware," *Patterns and Skeletons for Parallel and Distributed Computing*, edited by Fethi Rabhi and Sergei Gorlatch, Springer Verlag, 2002.

BC30 Richard E. Schantz and Douglas C. Schmidt, "Middleware for Distributed Systems: Evolving the Common Structure for Network-centric Applications," *Encyclopedia of Software Engineering*, edited by John Marciniak and George Telecki, Wiley and Sons, 2001.

BC29 Sumedh Mungee, Nagarajan Surendran, Yamuna Krishnamurthy, and Douglas C. Schmidt "The Design and Performance of a CORBA Audio/Video Streaming Service," *Multimedia Networking: Technology, Management, and Applications*, edited by Mahbubur Syed, Idea Group Publishing, Hershey, USA, 2001.

BC28 Nanbor Wang, Douglas C. Schmidt, and Carlos O'Ryan "An Overview of the CORBA Component Model," *Component-Based Software Engineering*, (George Heineman and Bill Councill, eds.), Addison-Wesley, Reading, MA, 2001.

BC27 Douglas C. Schmidt, "Applying a Pattern Language to Develop Application-Level Gateways," Design Patterns in Communications, (Linda Rising, ed.), Cambridge University Press, 2000.

BC26 Douglas C. Schmidt and Chris Cleeland, "Applying a Pattern Language to Develop Extensible ORB Middleware," Design Patterns in Communications, (Linda Rising, ed.), Cambridge University Press, 2000.

BC25 Carlos O'Ryan, Fred Kuhns, Douglas C. Schmidt, and Jeff Parsons, "Applying Patterns to Develop a Pluggable Protocols Framework for Object Request Broker Middleware," Design Patterns in Communications, (Linda Rising, ed.), Cambridge University Press, 2000.

BC24 David L. Levine, Christopher D. Gill, and Douglas C. Schmidt, "Object Lifecycle Manager – A Complementary Pattern for Controlling Object Creation and Destruction," Design Patterns in Communications, (Linda Rising, ed.), Cambridge University Press, 2000.

BC23 Douglas C. Schmidt, "A Family of Design Patterns For Flexibly Configuring Network Services in Distributed Systems," Design Patterns in Communications, (Linda Rising, ed.), Cambridge University Press, 2000.

BC22 James Hu and Douglas C. Schmidt, "JAWS: A Framework for High-performance Web Servers," *Object-Oriented Application Frameworks* book, John Wiley & Sons, October, 1999.

BC21 Chris Cleeland and Douglas C. Schmidt, "External Polymorphism, An Object Structural Pattern for Transparently Extending C++ Concrete Data Types" in C++ Gems II, (Robert Martin, ed.), SIGS, NY, 1999.

BC20 Douglas C. Schmidt, "GPERF: A Perfect Hash Function Generator" in C++ Gems II, (Robert Martin, ed.), SIGS, NY, 1999.

BC19 Douglas C. Schmidt, Tim H. Harrison, and Nat Pryce, "Thread-specific Storage: an Object Behavioral Pattern for Efficiently Accessing per-Thread State" in C++ Gems II, (Robert Martin, ed.), SIGS, NY, 1999.

BC18 Irfan Pyarali, Tim Harrison, Douglas C. Schmidt, and Thomas Jordan, "Proactor: an Object Behavioral Pattern for Demultiplexing and Dispatching Handlers for Asynchronous Events," *Pattern Languages of Program Design,* (Harrison, Foote, and Rohnert, eds.), Addison-Wesley, Reading, MA, 1999.

BC17 Douglas C. Schmidt and Paul Stephenson, "Using Design Patterns to Evolve System Software from UNIX to Windows NT," In *The Patterns Handbook*, (Linda Rising, ed.), Cambridge University Press, 1998.

BC16 Douglas C. Schmidt, David L. Levine, and Chris Cleeland, "Architectures and Patterns for High-performance, Real-time CORBA Object Request Brokers," *Advances in Computers*, Academic Press, Ed., Marvin Zelkowitz, Volume 48, July 1999.

BC15 Douglas C. Schmidt and Tatsuya Suda, "A Framework for Measuring the Performance of Alternative Process Architectures for Parallel Communication Subsystems," in *Network Performance Modeling and Simulation*, Walrand, Bagchi, and Zobrist, Ed., Gordon and Breach Publishers, 1998.

BC14 Douglas C. Schmidt, "Applying Design Patterns and Frameworks to Develop Object-Oriented Communication Software," The Handbook of Programming Languages, Volume I, edited by Peter Salus, MacMillan Computer Publishing, 1997.

BC13 Chris Cleeland, Douglas C. Schmidt, and Tim H. Harrison, "External Polymorphism – An Object Structural Pattern for Transparently Extending Concrete Data Types," *Pattern Languages of Program Design,* (Martin, Buschmann, and Riehl, eds.), Addison-Wesley, Reading, MA, 1997.

BC12 Timothy H. Harrison, Douglas C. Schmidt, and Irfan Pyarali, "Asynchronous Completion Token – An Object Behavioral Pattern for Efficient Asynchronous Event Handling," *Pattern Languages of Program Design,* (Martin, Buschmann, and Riehl, eds.), Addison-Wesley, Reading, MA, 1997.

BC11 Douglas C. Schmidt and Timothy H. Harrison, "Double-Checked Locking – A Optimization Pattern for Efficiently Initializing and Accessing Thread-safe Objects," *Pattern Languages of Program Design,* (Martin, Buschmann, and Riehl, eds.), Addison-Wesley, Reading, MA, 1997.

BC10 Douglas C. Schmidt, "Acceptor and Connector – A Family of Object Creational Patterns for Initializing Communication Services," *Pattern Languages of Program Design,* (Martin, Buschmann, and Riehl, eds.), Addison-Wesley, Reading, MA, 1997.

BC9 Douglas C. Schmidt and Paul Stephenson, "Using Design Patterns to Evolve System Software from UNIX to Windows NT," In *Wisdom of the Gurus*, (Charles Bowman, ed.), Cambridge University Press, 1996.

BC8 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Distributed Programming Techniques" in *Wisdom of the Gurus*, (Charles Bowman, ed.), Cambridge University Press, 1996.

BC7 Douglas C. Schmidt, "A Case Study in C++ Design Evolution" in *C++ Gems,* (Stanley Lippman, ed.), SIGS, NY, 1996, pp. 99-120.

BC6 Douglas C. Schmidt and Steve Vinoski, "Distributed Object Computing in C++" in *C++ Gems,* (Stanley Lippman, ed.), SIGS, NY, 1996, pp. 303-316.

BC5 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Distributed Programming Techniques" in *C++ Gems,* (Stanley Lippman, ed.), SIGS, NY, 1996, pp. 316-336.

BC4 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Server Programming Techniques" in *C++ Gems,* (Stanley Lippman, ed.), SIGS, NY, 1996, pp. 337-362.

BC3 Douglas C. Schmidt and Charles D. Cranor, "Half-Sync/Half-Async: A Architectural Pattern for Efficient and Well-structured Concurrent I/O" in *Pattern Languages of Program Design,* (Coplien, Vlissides, and Kerth, eds.), Addison-Wesley, Reading, MA, 1996.

BC2 R. Greg Lavender and Douglas C. Schmidt, "Active Object: An Object Behavioral Pattern for Concurrent Programming," in *Pattern Languages of Program Design,* (Coplien, Vlissides, and Kerth, eds.), Addison-Wesley, Reading, MA, 1996.

BC1 Douglas C. Schmidt, "Reactor: An Object Behavioral Pattern for Concurrent Event Demultiplexing and Event Handler Dispatching," *Pattern Languages of Program Design,* (Addison-Wesley, 1995), edited by James O. Coplien and Douglas C. Schmidt.

- **Refereed Conference Publications**

C195 Peng Zhang, Douglas C. Schmidt, and Jules White, "A Pattern Sequence for Designing Blockchain-Based Healthcare Information Technology Systems," the 26th Pattern Languages of Programming conference, October 7-10, 2019, Ottawa, Ontario, Canada.

C194 Nick Guertin, Douglas C. Schmidt, and Harry Levinson, "All Capability to All Boats, All the Time: Adding Options to the TI/APB Process," Proceedings of the 2018 Joint Undersea Warfare Technology Fall Conference, Groton, CT, September 18th, 2018.

C193 Zhongwei Teng, Peng Zhang, Xiao Li, William Nock, Marcelino Rodriguez-Cancio, Denis

Gilmore, Jules White, Douglas C. Schmidt, and Jonathan C. Nesbitt, "Authentication and Usability in mHealth Apps," proceedings of the 2018 IEEE International Conference on E-health Networking, Application and Services (Healthcom), 17-20 September 2018, Ostrava, Czech Republic (winner of the "Outstanding Paper" award).

C192 Peng Zhang, Douglas C. Schmidt, Jules White, and Shelagh A. Mulvaney, Towards Precision Behavioral Medicine with IoT: Iterative Design and Optimization of a Self-Management Tool for Type 1 Diabetes," proceedings of the 2018 IEEE International Conference on Healthcare Informatics (ICHI 2018), New York, NY, USA, June 4-7, 2018.

C191 Nick Guertin, Douglas C. Schmidt, and Bill Scherlis, "Capability Composition and Data Interoperability to Achieve More Effective Results than DoD System-of-Systems Strategies," proceedings of 15th Annual Acquisition Research Symposium, May 9-10, 2018 in Monterey, CA.

C190 Peng Zhang, Douglas C. Schmidt, Jules White, and Gunther Lenz, "Metrics for Assessing Blockchain-based Healthcare Decentralized Apps," Proceedings of the IEEE Healthcom 2017, October 12-15, 2017, Dalian, China.

C189 Peng Zhang, Jules White, Douglas C. Schmidt, and Gunther Lenz, "Design of Blockchain-Based Apps Using Familiar Software Patterns to Address Interoperability Challenges in Healthcare," the 24th Pattern Languages of Programming (PLoP) conference, October 22-25, 2017, Vancouver, Canada.

C188 Aron Laszka, Michael Walker, Abhishek Dubey and Douglas Schmidt, "Providing Privacy, Safety, and Security in IoT-Based Transactive Energy Systems using Distributed Ledgers," The 7th International Conference on the Internet of Things (IoT 2017), October 22-25, 2017, Linz, Austria.

C187 Fangzhou Sun, Peng Zhang, Jules White, Douglas C. Schmidt, Jacob Staples, and Lee Krause, A Feasibility Study of Autonomically Detecting In-process Cyber-Attacks, Proceedings of the 3rd IEEE International Conference on Cybernetics (CYBCONF-2017), Special Session on Cyber Security.

C186 Peng Zhang, Jules White, Douglas C. Schmidt, and Tom Dennis, "Discussions of a Preliminary Hand Hygiene Compliance Monitoring Application-as-a-Service", 10th International Conference on Health Informatics - HEALTHINF 2017, 21 - 23 February, 2017, Porto, Portugal.

C185 Peng Zhang, Jules White, Douglas C. Schmidt, and Tom Dennis, "Applying Machine Learning Methods to Predict Hand Hygiene Compliance Characteristics," Proceedings of the Biomedical and Health Informatics Conference, Orlando, Florida, February 16-19, 2017.

C184 Subhav Pradhan, Shweta Khare, Fangzhou Sun, Abhishek Dubey, Janos Sallai, Aniruddha Gokhale, Douglas Schmidt, Martin Lehofer, and Monika Sturm, "Towards a Distributed and Resilient Platform for Smart City Systems," First IEEE/ACM Symposium on Edge Computing, October 27-28, 2016, Washington DC.

C183 Peng Zhang, Jules White, and Douglas C. Schmidt, "HoliCoW: Automatically Breaking Team-based Software Projects to Motivate Student Testing," Proceedings to the Software Engineering Education and Training track at the 38th International Conference on Software Engineering Austin, TX, May 14 - 22, 2016.

C182 Nickolas H. Guertin, Robert Sweeney, and Douglas C. Schmidt, "How the Navy Can Use Open Systems Architecture to Revolutionize Capability Acquisition," Naval Postgraduate School's Acquisition Research Symposium, May 13th 2015, Monterey, CA.

C181 Nickolas H. Guertin, Robert Sweeney, and Douglas C. Schmidt, "Benefits of Applying an Open Systems Architecture Approach," 17th Systems Engineering Conference, National Defense Industry Association, October 27-30, 2014, Washington DC.

C180 Kyoungho An, Aniruddha Gokhale, Sumant Tambe, Gerardo Pardo-Castellote, and Douglas C. Schmidt, "Content-based Filtering Discovery Protocol (CFDP): Scalable and Efficient OMG DDS Discovery Protocol," 8th ACM International Conference on Distributed Event-Based Systems, Mumbai, India, May 26-29, 2014.

C179 Balakrishnan Dasarathy, Kevin Sullivan, Douglas C. Schmidt, Douglas H. Fisher, and Adam Porter, "The Past, Present, and Future of MOOCs and Their Relevance to Software Engineering," 36th ACM/IEEE International Conference on Software Engineering, Hyderabad, India, May 31 - June 7th, 2014.

C178 Rick Leathart, Adam Porter, Douglas Schmidt, Michael O'Hare, Harry Crisp, and Barry Laird, "Capability-Based Technical Reference Frameworks for Open System Architecture Implementations," Systems Engineering Conference (SEDC) 20134, Washington DC, April 3rd and 4th, 2014, Chantilly, VA.

C177 Douglas C. Schmidt, Chris Gill, and Jules White, "Elastic Infrastructure to Support Computing Clouds for Large-scale Cyber-Physical Systems," Proceedings of the International Symposium on Object-Oriented Real-time Distributed Computing (ISORC), June 2014, Reno, Nevada.

C176 Douglas C. Schmidt and Zach McCormick, "Creating and Teaching a MOOC on Pattern-Oriented Software Architecture for Concurrent and Networked Software," Proceedings of the WaveFront Forum at the SPLASH 2013 conference, October 2013, Indianapolis, IN.

C175 Zach McCormick and Douglas C. Schmidt, "Data Synchronization Patterns in Mobile Application Design," Proceedings of the Pattern Languages of Programs (PLoP) 2012 conference, October 19-21, 2012, Tucson, Arizona.

C174 James Edmondson, Aniruddha Gokhale and Douglas Schmidt, "Approximation Techniques for Maintaining Real-time Deployments Informed by User-provided Dataflows Within a Cloud," 31st International Symposium on Reliable Distributed Systems (SRDS 2012), 8th-11th October 2012, Irvine, California.

C173 James Edmondson, Douglas C. Schmidt, and Aniruddha Gokhale "QoS-enabled Distributed Mutual Exclusion in Public Clouds," Proceedings of the 1st International Symposium on Secure Virtual Infrastructures (DOA-SVI'11), October 17-19, 2011, Crete, Greece.

C172 Brian Dougherty, Jules White, Russell Kegley, Jonathan Preston, Douglas C. Schmidt, and Aniruddha Gokhale, "Optimizing Integrated Application Performance with Cache-aware Metascheduling," Proceedings of the 1st International Symposium on Secure Virtual Infrastructures (DOA-SVI'11), October 17-19, 2011, Crete, Greece.

C171 Akram Hakiri, Aniruddha Gokhale, Douglas C. Schmidt, Berthou Pascal, Joe Hoffert, and Gayraud Thierry, "A SIP-based Network QoS Provisioning Framework for Cloud-hosted DDS Applications," Proceedings of the 1st International Symposium on Secure Virtual Infrastructures (DOA-SVI'11), October 17-19, 2011, Crete, Greece.

C170 James Hill and Douglas C. Schmidt, "Experiences with Service-Oriented Middleware for Dynamic Instrumentation of Enterprise Distributed Real-time and Embedded Systems," Proceedings of the 1st International Symposium on Secure Virtual Infrastructures (DOA-SVI'11), October 17-19, 2011, Crete, Greece.

C169 Will Otte, Aniruddha Gokhale, Douglas C. Schmidt, and Johnny Willemsen, "Infrastructure for Component-based DDS Application Development," proceedings of the Tenth International Conference on Generative Programming and Component Engineering (GPCE'11), October 22-23, 2011 Portland, Oregon, USA.

C168 Will Otte, Aniruddha Gokhale, and Douglas C. Schmidt, Techniques for Predictable Deployment Latencies in Large-scale Component-based Distributed Real-time and Embedded Systems, Proceedings of the 14th International ACM SIGSOFT Symposium on Component Based Software Engineering (CBSE-2011), June 21th - 23th, 2011, in Boulder, Colorado, USA.

C167 Chris Thompson, Hamilton Turner, Jules White, and Douglas C. Schmidt, Analyzing Mobile Application Software Power Consumption via Model-Driven Engineering, Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems, Algarve, Portugal, March 5-7, 2011.

C166 Joe Hoffert, Douglas C. Schmidt, Aniruddha Gokhale, "Adapting Distributed Real-time and Embedded Publish/Subscribe Middleware for Cloud-Computing Environments," Proceedings of the ACM/IFIP/USENIX 11th International Middleware Conference, Bangalore, India, November 30-Dec 3, 2010.

C165 Joe Hoffert and Douglas Schmidt, "Evaluating Supervised Machine Learning for Adapting Enterprise DRE Systems," Proceedings of the International Symposium on Intelligence Information Processing and Trusted Computing (IPTC 2010), Huanggang City, China, October 28-29, 2010.

C164 J. Benjamin Gotow, Krzysztof Zienkiewicz, Jules White, and Douglas C. Schmidt, "Addressing Challenges in Delivering Augmented Reality Applications to Smartphones," Proceedings of the

Third International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2010), June 30-July 2, 2010, Chicago, IL.

C163 Chris Thompson, Jules White, Brian Dougherty, Adam Albright, and Douglas C. Schmidt, "Using Smartphones and Wireless Mobile Networks to Detect Car Accidents and Provide Situational Awareness to Emergency Responders," Proceedings of the Third International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2010), June 30-July 2, 2010, Chicago, IL.

C162 James H. Hill, Hunt Sutherland, Douglas C. Schmidt, Thomas Silveria, John M. Slaby, Paul Staudinger, and Nikita A. Visnevski, "OASIS: A Service-Oriented Architecture for Dynamic Instrumentation of Enterprise Distributed Real-time and Embedded Systems," Proceedings of the 13th International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC '10), May 5-6, 2010, Carmona, Spain.

C161 Joseph P. Loyall, Matthew Gillen, Aaron Paulos, Larry Bunch, Marco Carvalho, James Edmondson, Pooja Varshneya, Douglas C. Schmidt, Andrew Martignoni, "Dynamic Policy-Driven Quality of Service in Service-Oriented Systems," Proceedings of the 13th International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC '10), May 5-6, 2010, Carmona, Spain.

C160 Jaiganesh Balasubramanian, Aniruddha Gokhale, Abhishek Dubey, Friedhelm Wolf, Chenyang Lu, Chris Gill, and Douglas C. Schmidt, "Middleware for Resource-Aware Deployment and Configuration of Fault-tolerant Real-time Systems," Proceedings of the 16th IEEE Real-Time and Embedded Technology and Applications Symposium Stockholm, Sweden, April 12 - 15, 2010.

C159 John S. Kinnebrew, Daniel L. C. Mack, Gautam Biswas, Douglas C. Schmidt, "Coordination of Planning and Scheduling Techniques for a Distributed, Multi-level, Multi-agent System", Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART), Valencia, Spain, January 22-24, 2010.

C158 Nilabja Roy, Yuan Xue, Aniruddha Gokhale, Larry Dowdy and Douglas C. Schmidt, "A Component Assignment Framework for Improved Capacity and Assured Performance inWeb Portals," Proceedings of the 11th International Symposium on Distributed Objects, Middleware, and Applications (DOA'09) Vilamoura, Algarve-Portugal, Nov 01 - 03, 2009.

C157 Joe Hoffert, Douglas C. Schmidt, and Aniruddha Gokhale, "Evaluating Transport Protocols for Real-time Event Stream Processing Middleware and Applications," Proceedings of the 11th International Symposium on Distributed Objects, Middleware, and Applications (DOA'09) Vilamoura, Algarve-Portugal, Nov 01 - 03, 2009.

C156 Joe Hoffert and Douglas C. Schmidt, "Maintaining QoS for Publish/Subscribe Middleware in Dynamic Environments," Fast Abstract, 3rd ACM International Conference on Distributed Event-Based Systems (DEBS 2009) July 6-9, 2009 - Nashville, TN, USA.

C155 Nilabja Roy, Larry Dowdy, and Douglas C. Schmidt, "The Impact of Variability on Soft Real-Time System Scheduling," Proceedings of the 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2009), Beijing, China, August 24-26, 2009.

C154 Friedhelm Wolf, Jaiganesh Balasubramanian, Aniruddha Gokhale, and Douglas C. Schmidt, "Component Replication based on Failover Units," Proceedings of the 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2009), Beijing, China, August 24-26, 2009.

C153 Jules White, David Benavides, Brian Dougherty, Douglas C. Schmidt, "Automated Reasoning for Multi-step Software Product-line Configuration Problems," Proceedings of the 13th International Software Product Line Conference (SPLC 2009) August 24-28, 2009, San Francisco, CA.

C152 Joseph Loyall, Marco Carvalho, Douglas Schmidt, Matthew Gillen, Andrew Martignoni III, Larry Bunch, James Edmondson, and David Corman, "QoS Enabled Dissemination of Managed Information Objects in a Publish-Subscribe-Query Information Broker," the SPIE Defense Transformation and Net-Centric Systems conference, April, 2009, Orlando, FL.

C151 Brian Dougherty, Jules White, Jaiganesh Balasubramanian, Chris Thompson, and Douglas C. Schmidt, "Deployment Automation with BLITZ," Proceedings of the Emerging Results

track at the 31st International Conference on Software Engineering, Vancouver, Canada, May 16-24, 2009.

C150 Brian Dougherty, Jules White, Chris Thompson, and Douglas C. Schmidt, "Automating Hardware and Software Evolution Analysis," Proceedings of the 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), April 13-16, 2009, San Francisco, CA USA.

C149 James H. Hill, Hamilton A. Turner, James R. Edmondson, and Douglas C. Schmidt, "Unit Testing Non-functional Concerns of Component-based Distributed Systems," Proceedings of the 2nd International Conference on Software Testing, Verification, and Validation, April 1 - 4, 2009, Denver, Colorado.

C148 John S. Kinnebrew, William R. Otte, Nishanth Shankaran, Gautam Biswas, and Douglas C. Schmidt, "Intelligent Resource Management and Dynamic Adaptation in a Distributed Real-time and Embedded Sensor Web System," Proceedings of the 12th International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC '09), Tokyo, Japan, March 17-20, 2009.

C147 Jaiganesh Balasubramanian, Sumant Tambe, Chenyang Lu, Aniruddha Gokhale, Christopher Gill, and Douglas C. Schmidt, "Adaptive Failover for Real-time Middleware with Passive Replication, Proceedings of the 15th Real-time and Embedded Applications Symposium (RTAS) 2009, San Francisco, CA, United States, April 13 - 16, 2009.

C146 William R. Otte, John. S. Kinnebrew, Douglas C. Schmidt, and Gautam Biswas, "A Flexible Infrastructure for Distributed Deployment in Adaptive Sensor Webs," Proceedings of the 2009 IEEE Aerospace Conference, Big Sky, Montana, March 2009.

C145 Nanbor Wang, Douglas C. Schmidt, Angelo Corsaro, and Hans Van'T Hag, "Toward an Adaptive Data Distribution Service for Dynamic Large-Scale Network-Centric Operation and Warfare Systems," Proceedings of the 2008 Military Communications Conference, November 17-19, 2008 in San Diego, CA.

C144 Joe Hoffert, Douglas C. Schmidt, and Aniruddha Gokhale, "DQML: A Modeling Language for Configuring Publish/Subscribe Quality of Service Policies," Proceedings of the Distributed Objects, Middleware, and Applications (DOA'08), Monterrey, Mexico, Nov 10 - 12, 2008.

C143 Nilabja Roy, Akshay Dabholkar, Nathan Hamm, Larry Dowdy and Douglas Schmidt, "Modeling Software Contention using Colored Petri Nets," Proceedings of the 16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), September 8-10 2008, Baltimore, MD.

C142 Jules White, Douglas C. Schmidt, David Benavides, Pablo Trinidad, Antonio Ruiz-Cortez, "Automated Diagnosis of Product-line Configuration Errors in Feature Models," Proceedings of the Software Product Lines Conference (SPLC), September, 2008, Limerick, Ireland.

C141 Jules White and Douglas C. Schmidt, "Model-Driven Product-Line Architectures for Mobile Devices," Proceedings of the 17th Annual Conference of the International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008.

C140 Jules White and Douglas C. Schmidt, "Automated Configuration of Component-based Distributed Real-time and Embedded Systems from Feature Models," Proceedings of the 17th Annual Conference of the International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008.

C139 William R. Otte, John. S. Kinnebrew, Douglas C. Schmidt, Gautam Biswas, and Dipa Suri, "Application of Middleware and Agent Technologies to a Representative Sensor Network," Proceedings of the 2008 Earth Science Technology Conference, June 24-26, 2008, University of Maryland.

C138 Jai Balasubramanian, Aniruddha Gokhale, Douglas C. Schmidt, and Nanbor Wang, "Towards Middleware for Fault-tolerance in Distributed Real-time and Embedded Systems," Proceedings of the 8th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2008), Oslo, Norway, June 4-6, 2008.

C137 Serena Fritsch, Aline Senart, Douglas C. Schmidt, and Siobhan Clarke, "Time-bounded Adaptation for Automotive System Software," Proceedings of the Experience Track on Automotive Systems at the 30th International Conference on Software Engineering, Leipzig, Germany, 10 - 18 May 2008.

C136 Nilabja Roy, John S. Kinnebrew, Nishanth Shankaran, Gautam Biswas, and Douglas C. Schmidt, "Toward Effective Multi-capacity Resource Allocation in Distributed Real-time and Embedded Systems," Proceedings of the 11th IEEE International Symposium on Object/-Component/Service-oriented Real-time Distributed Computing, Orlando, Florida, May 5-7, 2008.

C135 Gan Deng, Douglas C. Schmidt, and Aniruddha Gokhale, "CaDANCE: A Criticality-Aware Deployment And Configuration Engine," Proceedings of the 11th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing, Orlando, Florida, May 5-7, 2008.

C134 Krishnakumar Balasubramanian and Douglas C. Schmidt, "Physical Assembly Mapper: A Model-driven Optimization Tool for QoS-enabled Component Middleware," Proceedings of 14th IEEE Real-Time and Embedded Technology and Applications Symposium, St. Louis, MO, United States, April 22 - April 24, 2008.

C133 Jaiganesh Balasubramanian, Sumant Tambe, Balakrishnan Dasarathy, Shrirang Gadgil, Frederick Porter, Aniruddha Gokhale, and Douglas C. Schmidt, "NetQoPE: A Model-driven Network QoS Provisioning Engine for Distributed Real-time and Embedded Systems," Proceedings of 14th IEEE Real-Time and Embedded Technology and Applications Symposium, St. Louis, MO, United States, April 22 - April 24, 2008.

C132 James Hill, Douglas C. Schmidt, John Slaby, and Adam Porter, "CiCUTS: Combining System Execution Modeling Tools with Continuous Integration Environments," Proceedings of the 15th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS), March 31st - 4th April, 2008 Belfast, Northern Ireland.

C131 Vinny Cahill, Aline Senart, Douglas C. Schmidt, Stefan Weber, Anthony Harrington, Barbara Hughes, and Kulpreet Singh, "The Managed Motorway: Real-time Vehicle Scheduling: A Research Agenda," Proceedings of the ACM HotMobile 2008, Silverado Resort, Napa Valley, CA, USA February 25-26, 2008.

C130 Jules White, Krzysztof Czarnecki, Douglas Schmidt, Gunther Lenz, Christoph Wienands, Egon Wuchner, Ludger Fiege, "Automated Model-based Configuration of Enterprise Java Applications," The 11th IEEE International EDOC Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland U.S.A.

C129 Joe Hoffert, Douglas Schmidt, and Aniruddha Gokhale, "A QoS Policy Configuration Modeling Language for Publish/Subscribe Middleware Platforms," Proceedings of International Conference on Distributed Event-Based Systems (DEBS), June 20-22nd, 2007, Toronto, Canada.

C128 Shanshan Jiang, Yuan Xue, and Douglas Schmidt, "Minimum Disruption Service Composition and Recovery in Mobile Ad hoc Networks, Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2007), August 6-10, 2007 - Philadelphia, PA.

C127 Jules White, Andrey Nechypurenko, Egon Wuchner, and Douglas C. Schmidt, "Optimizing and Automating Product-Line Variant Selection for Mobile Devices," Proceedings of the 11th International Software Product Line Conference, Kyoto, Japan, Sept 10-14, 2007.

C126 Nishanth Shankaran, Douglas C. Schmidt, Yingming Chen, Xenofon Koutsoukous, and Chenyang Lu, The Design and Performance of Configurable Component Middleware for End-to-End Adaptation of Distributed Real-time Embedded Systems, proceedings of the 10th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing, May 7-9, 2007, Santorini Island, Greece.

C125 Amogh Kavimandan, Krishnakumar Balasubramanian, Nishanth Shankaran, Aniruddha Gokhale, and Douglas C. Schmidt, QUICKER: A Model-driven QoS Mapping Tool, proceedings of the 10th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC), May 7-9, 2007, Santorini Island, Greece.

C124 Krishnakumar Balasubramanian, Douglas C. Schmidt, Zoltan Molnar, and Akos Ledeczi, Component-based System Integration via (Meta)Model Composition, Proceedings of the 14th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), March 26th-29th, 2007, Tucson, Arizona.

C123 Joe Hoffert, Shanshan Jiang, and Douglas C. Schmidt, "A Taxonomy of Discovery Services and Gap Analysis for Ultra-Large Scale Systems," Proceedings of the ACMSE 2007: 45th ACM Southeast Conference Winston-Salem, North Carolina, USA March 23-24, 2007.

C122 Andrey Nechypurenko, Egon Wuchner, Jules White, and Douglas C. Schmidt, Application of Aspect-based Modeling and Weaving for Complexity Reduction in Development of Automotive Distributed Realtime Embedded System, Proceedings of the Sixth ACM International Conference on Aspect-Oriented Software Development, Vancouver, British Columbia, March 12-16, 2007.

C121 Dipa Suri, Adam Howell, Douglas C. Schmidt, Gautam Biswas, John Kinnebrew, Will Otte, and Nishanth Shankaran, "A Multi-agent Architecture for Smart Sensing in the NASA Sensor Web," Proceedings of the 2007 IEEE Aerospace Conference Big Sky, Montana, March 3-10, 2007.

C120 John S. Kinnebrew, Ankit Gupta, Nishanth Shankaran, Gautam Biswas, and Douglas C. Schmidt, A Decision-Theoretic Planner with Dynamic Component Reconfiguration for Distributed Real-Time Applications, Proceedings of the The 8th International Symposium on Autonomous Decentralized Systems (ISADS 2007) Sedona, Arizona, Wednesday March 21 - Friday March 23, 2007.

C119 Ming Xiong, Jeff Parsons, James Edmondson, and Douglas C. Schmidt, "Evaluating Technologies for Tactical Information Management in Net-Centric Systems, Proceedings of the Defense Transformation and Net-Centric Systems conference, April 9-13, 2007, Orlando, Florida.

C118 Nilabja Roy, Nishanth Shankaran, and Douglas C. Schmidt "Target Manager: A Resource Provisioning Service for Enterprise Distributed Real-time and Embedded Systems," Proceedings of the International Symposium on Distributed Objects and Applications (DOA), Montpellier, France, Oct 29 - Nov 3, 2006.

C117 James Hill, John Slaby, Steve Baker, and Douglas C. Schmidt, "Evaluating Enterprise Distributed Real-time and Embedded System Quality of Service with System Execution Modeling Tools," Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Sydney, Australia, 16-18 August 2006.

C116 Nishanth Shankaran, Xenofon Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Yuan Xue, "Hierarchical Control of Multiple Resources in Distributed Real-time and Embedded Systems," Proceedings of the 18th Euromicro Conference on Real-Time Systems (ECRTS 06), Dresden, Germany, July 5-7, 2006.

C115 Dipa Suri, Adam Howell, Nishanth Shankaran, John Kinnebrew, Will Otte, Doug Schmidt, and Gautam Biswas, "Onboard Processing using the Adaptive Network Architecture", Proceedings of the Sixth annual NASA Earth Science Technology Conference MD, June 27th - 29th, 2006, College Park, MD.

C114 Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale, and Andrey Nechypurenko, "Modularizing Variability and Scalability Concerns in Distributed Real-time and Embedded Systems with Modeling Tools and Component Middleware, Proceedings of the 9th IEEE International Symposium on Object-oriented Real-time Distributed Computing (ISORC '06), April 24-26, 2006, Gyeongju, Korea.

C113 Stoyan Paunov and Douglas C. Schmidt, "RepoMan: A Component Repository Manager for Enterprise Distributed Real-time and Embedded Systems", Proceedings of the 44th ACM Southeast Conference, Melbourne, FL, March 10-12, 2006.

C112 Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, and Venkatesh Prasad Ranganat, "Context-Specific Middleware Specialization Techniques for Optimizing Software Product-line Architectures," Proceedings of ACM EuroSys 2006, Leuven, Belgium, April 18-21, 2006.

C111 Stoyan Paunov, James Hill, Douglas C. Schmidt, John Slaby, and Steve Baker, "Domain-Specific Modeling Languages for Configuring and Evaluating Enterprise DRE System Quality of Service," Proceedings of the 13th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '06), March 27th-30th, 2006, University of Potsdam, Potsdam, Germany.

C110 Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, and Venkatesh Prasad Ranganat, "Towards Highly Optimized Real-time Middleware for Software Product-line Architectures," Proceedings of the Work-In-Progress session at the 26th IEEE Real-Time Systems Symposium, December 5-8, 2005, Miami, Florida.

C109 Gan Deng, Jaiganesh Balasubramanian, William Otte, Douglas C. Schmidt, and Aniruddha Gokhale, DAnCE: A QoS-enabled Component Deployment and Configuration Engine,

Proceedings of the 3rd Working Conference on Component Deployment, Grenoble, France, November 28-29, 2005, p. 67-82.

C108 Jaiganesh Balasubramanian, Balachandran Natarajan, Douglas C. Schmidt, Aniruddha Gokhale, Gan Deng, and Jeff Parsons, "Evaluating Techniques for Dynamic Component Updating," Proceedings of the International Symposium on Distributed Objects and Applications (DOA), Agia Napa, Cyprus, Oct 31 - Nov 4, 2005, p. 978-996.

C107 Jules White, Douglas Schmidt, and Aniruddha Gokhale, "Simplifying Autonomic Enterprise Java Bean Applications via Model-driven Development: a Case Study," Proceedings of MODELS 2005, ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Half Moon Resort, Montego Bay, Jamaica, October 5-7, 2005, p. 601-615.

C106 Arvind Krishna, Douglas C. Schmidt, and Michael Stal, "Context Object A Design Pattern for Efficient Middleware Request Processing," Proceedings of the 12th Pattern Language of Programming Conference, Allerton Park, Illinois, September 7-10, 2005.

C105 Michael Stal and Douglas C. Schmidt, "Activator," Proceedings of the 12th Pattern Language of Programming Conference, Allerton Park, Illinois, September 7-10, 2005.

C104 Jules White, Boris Kolpackov, Balachandran Natarajan, and Douglas C. Schmidt, "Reducing Code Complexity With Vocabulary-Specific XML Language Bindings," Proceedings of the 43nd ACM Southeastern conference in Atlanta, GA, March 2005.

C103 Cemal Yilmaz, Arvind Krishna, Atif Memon, Adam Porter, Douglas C. Schmidt, Aniruddha Gokhale, and Bala Natarajan, "A Distributed Continuous Quality Assurance Process for Monitoring Performance Degradation in Evolving Software Systems, proceedings of the 27th International Conference on Software Engineering, St. Louis, MO, May 15-21, 2005, p. 293-302.

C102 Arvind S. Krishna, Emre Turkay, Aniruddha Gokhale, and Douglas C. Schmidt, Model-Driven Techniques for Evaluating the QoS of Middleware Configurations for DRE Systems, Proceedings of the 11th IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco, CA, March 2005, p. 180-189.

C101 Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Jeff Parsons, Aniruddha Gokhale, and Douglas C. Schmidt, "A Platform-Independent Component Modeling Language for Distributed Real-time and Embedded Systems," Proceedings of the 11th IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco, CA, March 2005, p. 190-199.

C100 Nanbor Wang, Christopher Gill, Douglas C. Schmidt, and Venkita Subramonian, "Configuring Real-time Aspects in Component Middleware," Proceedings of the Conference on Distributed Objects and Applications (DOA 2004), October 25-29, 2004, Cyprus, Greece.

C99 Jaiganesh Balasubramanian, Douglas C. Schmidt, Lawrence Dowdy, and Ossama Othman, "Evaluating the Performance of Middleware Load Balancing Strategies," Proceedings of the 8th International IEEE Enterprise Distributed Object Computing Conference, Monterey, California, September 20-24, 2004.

C98 George Edwards, Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale, and Balachandran Natarajan, "Model-driven Configuration and Deployment of Component Middleware Publisher/Subscriber Services," Proceedings of the 3rd ACM International Conference on Generative Programming and Component Engineering, Vancouver, CA, October 2004.

C97 Andrey Nechypurenko, Douglas C. Schmidt, Tao Lu, Gan Deng, Emre Turkey, and Aniruddha Gokhale, "Concern-based Composition and Reuse of Distributed Systems," Proceedings of the 8th International Conference on Software Reuse, ACM/IEEE, Madrid, Spain, July 2004.

C96 Arvind Krishna, Douglas C. Schmidt, Adam Porter, Atif Memon, Diego Sevilla-Ruiz, "Improving the Quality of Performance-intensive Software via Model-integrated Distributed Continuous Quality Assurance," Proceedings of the 8th International Conference on Software Reuse, ACM/IEEE, Madrid, Spain, July 2004.

C95 Chris Gill, Jeanna M. Gossett, David Corman, Joseph P. Loyall, Richard E. Schantz, Michael Atighetchi, and Douglas C. Schmidt, "Integrated Adaptive QoS Management in Middleware: An Empirical Case Study," Proceedings of the 10th Real-time Technology and Application Symposium, May 25-28, 2004, Toronto, CA.

C94 Pradeep Gore, Douglas C. Schmidt, Chris Gill, and Irfan Pyarali, "The Design and Performance of a Real-time Notification Service," Proceedings of the 10th IEEE Real-time Technol-

ogy and Application Symposium (RTAS '04), Toronto, CA, May 2004.

C93 Yamuna Krishnamurthy, Chris Gill, Douglas C. Schmidt, Irfan Pyarali, Louis Mgeta, Yuanfang Zhang, and Stephen Torri, "The Design and Performance of Real-time CORBA 2.0: Dynamic Scheduling in TAO," Proceedings of the 10th IEEE Real-time Technology and Application Symposium (RTAS '04), Toronto, CA, May 2004.

C92 Arvind S. Krishna, Nanbor Wang, Balachandran Natarajan, Aniruddha Gokhale, Douglas C. Schmidt and Gautam Thaker, "CCMPerf: A Benchmarking Tool for CORBA Component Model Implementations," Proceedings of the 10th IEEE Real-time Technology and Application Symposium (RTAS '04), Toronto, CA, May 2004.

C91 George Edwards, Douglas C. Schmidt, Aniruddha Gokhale, and Bala Natarajan, Integrating Publisher/Subscriber Services in Component Middleware for Distributed Real-time and Embedded Systems, Proceedings of the 42nd ACM Southeastern conference in Huntsville, AL, April 2004.

C90 Atif Memon, Adam Porter, Cemal Yilmaz, Adithya Nagarajan, Douglas C. Schmidt, and Bala Natarajan, "Skoll: Distributed Continuous Quality Assurance," Proceedings of the 26th IEEE/ACM International Conference on Software Engineering, Edinburgh, Scotland, May 2004.

C89 Arvind Krishna, Douglas C. Schmidt, and Raymond Klefstad, "Enhancing Real-Time CORBA via Real-Time Java," Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS), March 23-26, 2004, Tokyo, Japan.

C88 Arvind Krishna, Douglas C. Schmidt, Krishna Raman, and Raymond Klefstad, "Enhancing Real-time CORBA Predictability and Performance," Proceedings of the 5th International Symposium on Distributed Objects and Applications (DOA), Catania, Sicily, November 2003.

C87 Jeff Gray, Ted Bapty, Sandeep Neema, Douglas C. Schmidt, Aniruddha Gokhale, and Balachandran Natarajan, "An Approach for Supporting Aspect-Oriented Domain Modeling, Proceedings of the 2nd Generative Programming and Component Engineering (GPCE '03) conference, Erfurt, Germany, September 22-25, 2003.

C86 Richard E. Schantz, Joseph P. Loyall, Douglas C. Schmidt, Craig Rodrigues, Yamuna Krishnamurthy, and Irfan Pyarali, "Flexible and Adaptive QoS Control for Distributed Real-time and Embedded Middleware," Proceedings of Middleware 2003, 4th IFIP/ACM/USENIX International Conference on Distributed Systems Platforms, June 16-20, 2003, Rio de Janeiro, Brazil.

C85 Arvind Krishna, Douglas C. Schmidt, Raymond Klefstad, and Angelo Corsaro, "Towards Predictable Real-time Java Object Request Brokers," Proceedings of the 9th IEEE Real-time/Embedded Technology and Applications Symposium (RTAS), Washington DC, May 28-30, 2003.

C84 Douglas C. Schmidt and Frank Buschmann, "Patterns, Frameworks, and Middleware: Their Synergistic Relationships," Proceedings of the IEEE/ACM International Conference on Software Engineering, Portland, Oregon, May 3-10, 2003.

C83 Radu Cornea, Nikil Dutt, Rajesh Gupta, Ingolf Krueger, Alex Nicolau, Douglas C. Schmidt, and Sandeep Shukla, "FORGE: A Framework for Optimization of Distributed Embedded Systems Software," International Parallel and Distributed Processing Symposium, Nice, France, April 22-26 2003.

C82 Raymond Klefstad, Sumita Rao, and Douglas C. Schmidt, "Design and Performance of a Dynamically Configurable, Messaging Protocols Framework for Real-time CORBA," Proceedings of the Distributed Object and Component-based Software Systems part of the Software Technology Track at the 36th Annual Hawaii International Conference on System Sciences, January 6–9, 2003, Big Island of Hawaii.

C81 Raymond Klefstad, Arvind S. Krishna, and Douglas C. Schmidt, "Design and Performance of a Modular Portable Object Adapter for Distributed, Real-Time, Embedded CORBA Applications," Proceedings of the Distributed Objects and Applications (DOA) conference, Irvine, CA, October/November, 2002.

C80 Chris Gill, Fred Kuhns, Douglas C. Schmidt, and Ron Cytron, "Empirical Differences Between COTS Middleware Scheduling Paradigms," Proceedings of the Distributed Objects and Applications (DOA) conference, Irvine, CA, October/November, 2002.

C79 Angelo Corsaro and Douglas C. Schmidt, "jRate: The Chameleonic Real-Time Java Implementation," Proceedings of the Distributed Objects and Applications (DOA) conference, Irvine, CA, October/November, 2002.

C78 Mayur Deshpande, Douglas C. Schmidt, Carlos O'Ryan, and Darrell Brunsch, "The Design and Performance of Asynchronous Method Handling for CORBA," Proceedings of the Distributed Objects and Applications (DOA) conference, Irvine, CA, October/November, 2002.

C77 Irfan Pyarali, Douglas C. Schmidt, and Ron Cytron, "Achieving End-to-End Predictability of the TAO Real-time CORBA ORB," Proceedings of the $8^{th}$ IEEE Real-Time Technology and Applications Symposium, San Jose, CA, September 2002.

C76 Angelo Corsaro and Douglas C. Schmidt, "Evaluating Real-Time Java Features and Performance for Real-time Embedded Systems," Proceedings of the $8^{th}$ IEEE Real-Time Technology and Applications Symposium, San Jose, CA, September 2002.

C75 Angelo Corsaro, Douglas C. Schmidt, Raymond Klefstad, and Carlos O'Ryan, "Virtual Component: a Design Pattern for Memory-Constrained Embedded Applications," Proceedings of the $9^{th}$ Annual Conference on the Pattern Languages of Programs, Monticello, Illinois, September, 2002.

C74 Joseph K. Cross and Douglas C. Schmidt, "Quality Connector: A Pattern Language for Provisioning and Managing Quality-Constrained Services in Distributed Real-time and Embedded Systems Proceedings of the 9th Annual Conference on the Pattern Languages of Programs, Monticello, Illinois, September, 2002.

C73 Richard Schantz, Franklin Webber, Partha Pal, Joseph Loyall, and Douglas C. Schmidt, "Protecting Applications Against Malice with Adaptive Middleware," Certification and Security in E-Services stream of the 17th IFIP World Computer Congress, Montreal, Canada, August 25-30, 2002.

c72 Richard Schantz and Douglas C. Schmidt, "Research Advances in Middleware for Distributed Systems: State of the Art," Computer Communications stream of the 17th IFIP World Computer Congress, Montreal, Canada, August 25-30, 2002.

C71 Raymond Klefstad, Douglas C. Schmidt, and Carlos O'Ryan, "Towards Highly Configurable Real-time Object Request Brokers," the IEEE International Symposium on Object-Oriented Real-time Distributed Computing (ISORC), Washington DC, April 29–May 1, 2002.

C70 Angelo Corsaro, Douglas C. Schmidt, Chris Gill, and Ron Cytron, "Formalizing Meta-Programming Techniques to Reconcile Heterogeneous Scheduling Policies in Open Distributed Real-Time Systems," Proceedings of the 3rd International Symposium on Distributed Objects and Applications, September 8-10, 2001, Rome, Italy.

C69 David A. Karr, Craig Rodrigues, Yamuna Krishnamurthy, Irfan Pyarali, and Douglas C. Schmidt "Application of the QuO Quality-of-Service Framework to a Distributed Video Application," Proceedings of the 3rd International Symposium on Distributed Objects and Applications, September 8-10, 2001, Rome, Italy.

C68 Nanbor Wang, Kirthika Parameswaran, and Douglas C. Schmidt, "The Design and Performance of Meta-Programming Mechanisms for Object Request Broker Middleware," Proceedings of the 6th USENIX Conference on Object-Oriented Technologies and Systems (COOTS), San Antonio, TX, Jan/Feb, 2001.

C67 Andy Gokhale, Bala Natarajan, Douglas C. Schmidt and Shalini Yajnik, "Applying Patterns to Improve the Performance of Fault-Tolerant CORBA," of the 7th International Conference on High Performance Computing (HiPC 2000), ACM/IEEE, Bangalore, India, December 2000.

C66 Nanbor Wang, Michael Kircher, and Douglas C. Schmidt, "Applying Reflective Techniques to Optimize a QoS-enabled CORBA Component Model Implementation, the 24th Annual International Computer Software and Applications Conference (COMPSAC 2000), Taipai, Taiwan, October 25-27 2000.

C65 Frederic Andres, Nicolas Dessaigne, Jose Martinez, Noureddine Mouaddib, Kinji Ono, Douglas C. Schmidt, Panrit Tosukhowong, "MISE: The MediaSys Image Search Engine," 11th International Conference on Database and Expert Systems Applications (DEXA 2000), London, UK, September 2000.

C64 Andy Gokhale, Bala Natarajan, Douglas C. Schmidt and Shalini Yajnik, "DOORS: Towards High-performance Fault-Tolerant CORBA," Proceedings of the 2nd International Symposium

on Distributed Objects and Applications (DOA '00), OMG, Antwerp, Belgium, September 2000.

C63 Irfan Pyarali, Carlos O'Ryan, Douglas C. Schmidt, "A Pattern Language for Efficient, Predictable, Scalable, and Flexible Dispatching Components," Proceedings of the 7th Pattern Language of Programming Conference, Monticello, Illinois, August, 2000.

C62 Douglas C. Schmidt, Carlos O'Ryan, Irfan Pyarali, Michael Kircher and Frank Buschmann, "Leader/Followers: A Design Pattern for Efficient Multi-threaded Event Demultiplexing and Dispatching," Proceedings of the 7th Pattern Languages of Programming Conference, Monticello, Illinois, August 2000.

C61 Carlos O'Ryan, Douglas C. Schmidt, Fred Kuhns, Marina Spivak, Jeff Parsons Irfan Pyarali, and David L. Levine, "Evaluating Policies and Mechanisms for Supporting Embedded, Real-Time Applications with CORBA 3.0," Proceedings of the Sixth IEEE Real-Time Technology and Applications Symposium (RTAS'00), Washington D.C., USA, May 31-June 2, 2000.

C60 Nanbor Wang, Douglas C. Schmidt, and David Levine, "Optimizing the CORBA Component Model for High-performance and Real-time Applications," Work-in-progress session of the IFIP/ACM Middleware 2000 Conference, Pallisades, New York, April 3-7, 2000.

C59 Alexander B. Arulanthu, Carlos O'Ryan, Douglas C. Schmidt, Michael Kircher, and Jeff Parsons, "The Design and Performance of a Scalable ORB Architecture for CORBA Asynchronous Messaging," Proceedings of the IFIP/ACM Middleware 2000 Conference, Pallisades, New York, April 3-7, 2000.

C58 Carlos O'Ryan, Fred Kuhns, Douglas C. Schmidt, Ossama Othman, and Jeff Parsons, The Design and Performance of a Pluggable Protocols Framework for Real-time Distributed Object Computing Middleware, Proceedings of the IFIP/ACM Middleware 2000 Conference, Pallisades, New York, April 3-7, 2000.

C57 Irfan Pyarali, Carlos O'Ryan, and Douglas C. Schmidt, "A Pattern Language for Efficient, Predictable, Scalable, and Flexible Dispatching Mechanisms for Distributed Object Computing Middleware," Proceedings of the IEEE/IFIP International Symposium on Object-Oriented Real-time Distributed Computing, March 15-17, 2000, Newport Beach, California.

C56 David Levine, Douglas C. Schmidt, and Sergio Flores-Gaitan, "An Empirical Evaluation of OS Support for Real-time CORBA Object Request Brokers," Proceedings of the Multimedia Computing and Networking 2000 (MMCN00) conference, ACM, San Jose, CA, January 25-27 2000.

C55 Douglas C. Schmidt, "Middleware Techniques and Optimizations for Real-time, Embedded Systems," Proceedings of the 12th International Symposium On System Synthesis, IEEE, San Jose, CA, USA November, 11, 1999.

C54 Panrit Tosukhowong, Frederic Andres, Kinji Ono, Nicolas Dessaigne, Josi Martinez, Nouredine Mouaddib, Douglas C. Schmidt, "A Flexible Image Search Engine," Proceedings of International Multimedia Conference Archive Proceedings of the Seventh ACM International Conference on Multimedia, Orlando, Florida, United States, November 1-5, 1999.

C53 Bryan S. Doerr, Thomas Venturella, Rakesh Jha, Christopher D. Gill, and Douglas C. Schmidt, "Adaptive Scheduling for Real-time, Embedded Information Systems," Proceedings of the 18th IEEE/AIAA Digital Avionics Systems Conference (DASC), St. Louis, Missouri, October 24-29, 1999.

C52 Christopher D. Gill, David L. Levine, Carlos O'Ryan, and Douglas C. Schmidt, "Distributed Object Visualization for Sensor-Driven Systems," Proceedings of the 18th IEEE/AIAA Digital Avionics Systems Conference (DASC), St. Louis, Missouri, October 24-29, 1999.

C51 Fred Kuhns, Douglas C. Schmidt, and David L. Levine, "The Performance of a Real-time I/O Subsystem for QoS-enabled ORB Middleware," Proceedings of the International Symposium on Distributed Objects and Applications (DOA '99), Edinburgh, Scotland, September 1999.

C50 David L. Levine, Christopher D. Gill, and Douglas C. Schmidt, "Object Lifecycle Manager – A Complementary Pattern for Controlling Object Creation and Destruction." Proceedings of the 5th Pattern Languages of Programing Conference, Allerton Park, Illinois, USA, 15–18 August 1999.

C49 Fred Kuhns, Douglas C. Schmidt, David Levine, and Rajeev Bector, "The Design and Performance of a Real-time I/O Subsystem," Proceedings of the 5th IEEE Real-Time Technology

and Applications Symposium (RTAS99), Vancouver, British Columbia, Canada, June 2-4, 1999.

C48 Irfan Pyarali, Carlos O'Ryan, Douglas C. Schmidt, Nanbor Wang, Vishal Kachroo, and Aniruddha Gokhale, "Applying Optimization Patterns to Design Real-time ORBs," Proceedings of the 5th USENIX Conference on Object-Oriented Technologies and Systems, May 3-7, 1999, San Diego, CA.

C47 Andy Gokhale and Douglas C. Schmidt, "Techniques for Optimizing CORBA Middleware for Distributed Embedded Systems" Proceedings of INFOCOM '99, March 21-25th, New York, New York.

C46 Sumedh Mungee, Nagarajan Surendran, and Douglas C. Schmidt, "The Design and Performance of a CORBA Audio/Video Streaming Service," Proceedings of the 31st Hawaii International Conference on System Systems (HICSS), Hawaii, January, 1999, minitrack on Multimedia DBMS and the WWW, Hawaii, January 1999.

C45 Chris D. Gill, David L. Levine, and Douglas C. Schmidt, "Dynamic Scheduling for Avionics Applications," Proceedings of the 17th IEEE/AIAA Digital Avionics System Conference, 31 October - 6 November 1998.

C44 James Hu, Irfan Pyarali, and Douglas C. Schmidt, "Applying the Proactor Pattern to High-Performance Web Servers," Proceedings of the 10th International Conference on Parallel and Distributed Computing and Systems, IASTED, Las Vegas, Nevada, October 28-31, 1998.

C43 Douglas C. Schmidt, Sumedh Mungee, and Andy Gokhale, "Alleviating Priority Inversion and Non-determinism in Real-time CORBA ORB Core Architectures," Proceedings of the Fourth IEEE Real-Time Technology and Applications Symposium (RTAS), Denver, Colorado, June 3-5, 1998.

C42 Prashant Jain, Seth Widoff, and Douglas C. Schmidt, "The Design and Performance of Med-Java, A Distributed Electronic Medical Imaging System Developed with Java Applets and Web Tools" Proceedings of the 4th USENIX Conference on Object-Oriented Technologies and Systems, Sante Fe, New Mexico, April 1998. This was selected as the best student paper in the conference.

C41 James Hu, Sumedh Mungee, and Douglas C. Schmidt, "Techniques for Developing and Measuring High-performance Web Servers over ATM Networks," Proceedings of INFOCOM '98, San Francisco, March/April, 1998.

C40 Aniruddha Gokhale and Douglas C. Schmidt, "Optimizing the Performance of the CORBA Internet Inter-ORB Protocol Over ATM," Proceedings of the 31st Hawaii International Conference on System Systems (HICSS), Hawaii, January, 1998. This was selected as the best paper in the Software Technology Track (188 submitted, 77 accepted).

C39 Aniruddha Gokhale and Douglas C. Schmidt, "Evaluating the Performance of Demultiplexing Strategies for Real-time CORBA," Proceedings of GLOBECOM '97 conference, IEEE, Phoenix, AZ, November, 1997.

C38 James Hu, Irfan Pyarali, and Douglas C. Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-speed Networks," Proceedings of the 2nd Global Internet Conference (held as part of GLOBECOM '97) in Phoenix, AZ, November 4-8, 1997.

C37 Tim Harrison and David Levine and Douglas C. Schmidt, "The Design and Performance of a Real-time CORBA Event Service," Proceedings of OOPSLA '97, ACM, Atlanta, GA, October 1997.

C36 Aniruddha Gokhale and Douglas C. Schmidt and Stan Moyer, "Tools for Automating the Migration from DCE to CORBA,"Proceedings of ISS 97: World Telecommunications Congress, IEEE Toronto, Canada, September, 1997.

C35 Jain and Douglas C. Schmidt, "Service Configurator – A Pattern for Dynamic Configuration of Services," the 4th annual Pattern Languages of Programmingy conference in Allerton Park, Illinois, September 1997.

C34 Chris Cleeland, Douglas C. Schmidt, and Tim H. Harrison, "External Polymorphism – An Object Structural Pattern for Transparently Extending C++ Concrete Data Types," the 4th annual Pattern Languages of Programmingy conference in Allerton Park, Illinois, September 1997.

C33 Douglas C. Schmidt, Tim H. Harrison, and Nat Pryce, "Thread-specific Storage: an Object Behavioral Pattern for Efficiently Accessing per-Thread State," The 4th annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 1997.

C32 Irfan Pyarali, Tim Harrison, Douglas C. Schmidt, and Thomas Jordan, "Proactor: an Object Behavioral Pattern for Demultiplexing and Dispatching Handlers for Asynchronous Events," the 4th annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 1997.

C31 Prashant Jain and Douglas C. Schmidt, "Service Configurator – A Pattern for Dynamic Configuration of Services," Proceedings of the 3rd Conference on Object-Oriented Technologies and Systems, USENIX, Portland, OR, June 16-19, 1997.

C30 Aniruddha Gokhale and Douglas C. Schmidt, "Evaluating Latency and Scalability of CORBA Over High-Speed ATM Networks," Proceedings of the International Conference on Distributed Computing Systems '97, IEEE, Baltimore, Maryland, May 27–30, 1997.

C29 Aniruddha Gokhale and Douglas C. Schmidt, "Performance of the CORBA Dynamic Invocation Interface and Internet Inter-ORB Protocol over High-Speed ATM Networks," Proceedings of GLOBECOM '96, IEEE, London England, November, 1996.

C28 Aniruddha Gokhale and Douglas C. Schmidt, "Measuring the Performance of Communication Middleware on High-Speed Networks," Proceedings of SIGCOMM '96, ACM, San Francisco, August 28-30th, 1996.

C27 Irfan Pyarali, Tim Harrison, and Douglas. C. Schmidt, "Design and Performance of an Object-Oriented Framework for High-Speed Electronic Medical Imaging," Proceedings of the 2nd Conference on Object-Oriented Technologies and Systems (COOTS), USENIX, Toronto, June 18-22, 1996.

C26 Douglas C. Schmidt, "A Family of Design Patterns For Flexibly Configuring Network Services in Distributed Systems," Proceedings of the International Conference on Configurable Distributed Systems, IEEE, Annapolis, Maryland, May 6-8, 1996.

C25 Douglas C. Schmidt "Using Design Patterns to Develop High-Performance Object-Oriented Communication Software Frameworks," Proceedings of the 8th Annual Software Technology Conference, Salt Lake City, Utah, April 21-26, 1996.

C24 Douglas C. Schmidt, Timothy H. Harrison, and Irfan Pyarali, "An Object-Oriented Framework for High-Performance Electronic Medical Imaging," Proceedings of the *Very High Resolution and Quality Imaging* mini-conference at the Symposium on Electronic Imaging in the International Symposia Photonics West 1996, SPIE, San Jose, California USA, January 27 - February 2, 1996.

C23 Douglas C. Schmidt and Charles D. Cranor, "Half-Sync/Half-Async: A Pattern for Efficient and Well-structured Concurrent I/O," *The 2nd Pattern Languages of Programs Conference* Monticello, Illinois, September 6-8, 1995.

C22 R. Greg Lavender and Douglas C. Schmidt, "Active Object: An Object Behavioral Pattern for Concurrent Programming," *The 2nd Pattern Languages of Programs Conference*, Monticello, Illinois, September 6-8, 1995.

C21 Guru Parulkar, Douglas C. Schmidt, and Jonathan S. Turner, "${}_aI_tP_m$: a Strategy for Integrating IP with ATM," the Symposium on Communications Architectures and Protocols (SIGCOMM), ACM, Cambridge, MA, August 30 to September 1, 1995.

C20 Douglas C. Schmidt, Tim Harrison, and Ehab Al-Shaer, "Object-Oriented Components for High-speed Network Programming," *Proceedings of the Conference on Object-Oriented Technologies (COOTS)*, USENIX, June 26-29, 1995 Monterey, California, USA, pp. 21-38.

C19 Douglas C. Schmidt and Paul Stephenson, "Experience Using Design Patterns to Evolve Communication Software Across Diverse OS Platforms," *Proceedings of the 9th European Conference on Object-Oriented Programming (ECOOP)*, ACM, Aarhus, Denmark, August, 1995.

C18 Douglas C. Schmidt and Tatsuya Suda, "Measuring the Performance of Parallel Message-based Process Architectures," *Proceedings of the INFOCOM Conference on Computer Communications*, IEEE, Boston, MA, April, 1995, pp. 624-633.

C17 Douglas C. Schmidt and Tatsuya Suda, "Experiences with an Object-Oriented Architecture for Developing Dynamically Extensible Network Management Software," *Proceedings of the Globecom Conference*, IEEE, San Francisco, California, November, 1994, pp. 1-7.

Douglas C. Schmidt 

C16 Douglas C. Schmidt and Paul Stephenson, "Achieving Reuse Through Design Patterns," *Proceedings of the 3rd Annual C++ World Conference*, SIGS, Austin, Texas, November 14-18, 1994.

C15 Douglas C. Schmidt, "Developing Object-Oriented Frameworks to Dynamically Configure Concurrent, Multi-service Network Daemons," *Proceedings of the 3rd Annual C++ World Conference*, SIGS, Austin, Texas, November 14-18, 1994.

C14 Douglas C. Schmidt, "Reactor: An Object Behavioral Pattern for Concurrent Event Demultiplexing and Dispatching," *The 1st Annual Conference on the Pattern Languages of Programs*, Monticello, Illinois, August, 1994, pp. 1-10.

C13 Douglas C. Schmidt, "The ADAPTIVE Communication Environment: An Object-Oriented Network Programming Toolkit for Developing Communication Software," *Proceedings of the 12th Annual Sun Users Group Conference*, SUG, San Francisco, June 16-17, 1994. This paper won the "best student paper" award at the conference.

C12 Douglas C. Schmidt, Burkhard Stiller, Tatsuya Suda, and Martina Zitterbart, "Configuring Function-based Communication Protocols for Distributed Applications," *Proceedings of the 8th International Working Conference on Upper Layer Protocols, Architectures, and Applications*, IFIP, Barcelona, Spain, June 1-3, 1994, pp. 361-376.

C11 Douglas C. Schmidt and Tatsuya Suda, "The ADAPTIVE Service Executive: An Object-Oriented Architecture for Configuring Concurrent Distributed Communication Systems," *Proceedings of the 8th International Working Conference on Upper Layer Protocols, Architectures, and Applications*, IFIP, Barcelona, Spain, June 1-3, 1994, pp. 163-178.

C10 Douglas C. Schmidt, "ASX: An Object-Oriented Framework for Developing Distributed Applications," *Proceedings of the 6th C++ Conference*, USENIX, Cambridge, Massachusetts, April, 1994, pp. 200-220.

C9 Douglas C. Schmidt, "The ADAPTIVE Communication Environment: Object-Oriented Network Programming Components for Developing Client/Server Applications," *Proceedings of the 11th Annual Sun Users Group Conference*, SUG, San Jose, December 7-9, 1993, pp. 214-225. This paper won the "best student paper" award at the conference.

C8 Douglas C. Schmidt and Paul Stephenson, "An Object-Oriented Framework for Developing Network Server Daemons," *Proceedings of the 2nd Annual C++ World Conference*, SIGS, Dallas, Texas, October 18-22, 1993, pp. 73-85.

C7 Douglas C. Schmidt, "Object-Oriented Techniques for Developing Extensible Network Servers," *Proceedings of the 2nd Annual C++ World Conference*, SIGS, Dallas, Texas, October 18-22, 1993.

C6 Douglas C. Schmidt, Burkhard Stiller, Tatsuya Suda, Ahmed Tantawy, and Martina Zitterbart, "Configuration Support for Flexible Function-Based Communication Systems," *Proceedings of the 18th Conference on Local Computer Networks*, IEEE, Minneapolis, Minnesota, September 20-22, 1993, pp. 369-378.

C5 Douglas C. Schmidt and Tatsuya Suda, "ADAPTIVE: a Framework for Experimenting with High-Performance Transport System Process Architectures," *Proceedings of the 2nd International Conference on Computer Communications and Networks*, ISCA, San Diego, California, June 28-30, 1993, pp. 1-8.

C4 Donald F. Box, Douglas C. Schmidt, and Tatsuya Suda, "ADAPTIVE: An Object-Oriented Framework for Flexible and Adaptive Communication Protocols," *Proceedings of the 4th Conference on High Performance Networking*, IFIP, Liege, Belgium, December 14-18, 1992, pp. 367-382.

C3 Douglas C. Schmidt, Donald F. Box, and Tatsuya Suda, "ADAPTIVE: A Flexible and Adaptive Transport System Architecture to Support Lightweight Protocols for Multimedia Applications on High-Speed Networks," *Proceedings of the 1st Symposium on High Performance Distributed Computing*, IEEE, Syracuse, New York, September 9-11, 1992, pp. 174-186.

C2 Richard W. Selby, Adam A. Porter, Douglas C. Schmidt, and James Berney, "Metric-Driven Analysis and Feedback Systems for Enabling Empirically Guided Software Development," *Proceedings of the 13th Annual International Conference on Software Engineering*, IEEE, Austin, Texas, May, 1991, pp. 430-443.

C1 Douglas C. Schmidt "GPERF: A Perfect Hash Function Generator," *Proceedings of the 2nd C++ Conference*, USENIX, San Francisco, California, April 9-11, 1990, pp. 87-102.

- **Refereed Workshop Publications**

W74 Nick Guertin, Douglas C. Schmidt, and Harry Levinson, "Business and Organizational Impacts for Modular Flexible Ships," Proceedings of the 2018 Design Sciences Series Workshop on Modular Adaptable Ships, Washington DC, November 14-15, 2018.

W73 Michael Walker, Abhishek Dubey, Aron Laszka, and Douglas C. Schmidt, "PlaTIBART: a Platform for Transactive IoT Blockchain Applications with Repeatable Testing," Proceedings of the ACM/IFIP/USENIX 4th Workshop on Middleware and Applications for the Internet of Things, December 2017, Las Vegas, USA.

W72 Abhishek Dubey, Subhav Pradhan, Douglas C. Schmidt, Sebnem Rusitschka, and Monika Sturm, "The Role of Context and Resilient Middleware in Next Generation Smart Grids," Proceedings of the 3rd Middleware for Context-Aware Applications in the IoT (M4IOT 2016) Workshop at the ACM/IFIP/USENIX Middleware 2016 Conference, Dec 12 - 16, 2016, Trento, Italy.

W71 Violetta Vylegzhanina, Douglas C. Schmidt, and Jules White, "Gaps and Future Directions in Mobile Security Research," Proceedings of the Third International Workshop on Mobile Development Lifecycle, Pittsburgh, PA, October 26th, 2015.

W70 Violetta Vylegzhanina, Douglas C. Schmidt, Pamela Hull, Janice S. Emerson, Meghan E. Quirk, and Shelagh Mulvaney, "Helping Children Eat Well Via Mobile Software Technologies," Proceedings of the Second International Workshop on Mobile Development Lifecycle, October 21st, 2015, Portland, OR.

W69 Jules White and Douglas C. Schmidt, "R&D Challenges and Emering Softwares for Multicore Deployment/Configuration Optimization," proceedings of the ACM Workshop on Future of Software Engineering Research (FoSER 2010), Santa Fe, NM, November 7-11, 2010.

W68 Will Otte, Douglas C. Schmidt, and Aniruddha Gokhale, "Towards an Adaptive Deployment and Configuration Framework for Component-based Distributed Systems," Proceedings of the 9th Workshop on Adaptive and Reflective Middleware (ARM 2010) November 27, 2010, Bangalore India, collocated with Middleware 2010.

W67 Jaiganesh Balasubramanian, Alexander Mintz, Andrew Kaplan, Grigory Vilkov, Artem Gleyzer, Antony Kaplan, Ron Guida, Pooja Varshneya and Douglas Schmidt, "Adaptive Parallel Computing for Large-scale Distributed and Parallel Applications," Proceedings of the Workshop on Data Dissemination for Large-scale Complex Critical Infrastructures (DD4LCCI), 27 April 2010, in conjunction with EDCC 2010, Valencia - Spain, April 28-30, 2010.

W66 Joe Hoffert, Douglas Schmidt, and Aniruddha Gokhale, "Adapting and Evaluating Distributed Real-time and Embedded Systems in Dynamic Environments," Proceedings of the Workshop on Data Dissemination for Large-scale Complex Critical Infrastructures (DD4LCCI), 27 April 2010, in conjunction with EDCC 2010, Valencia - Spain, April 28-30, 2010.

W65 Joe Hoffert, Dan Mack, and Douglas C. Schmidt, "Using Machine Learning to Maintain Pub/Sub System QoS in Dynamic Environments, Proceedings of the 8th Workshop on Adaptive and Reflective Middleware (ARM'09) December 1st 2009, Urbana Champaign, Illinois, USA collocated with Middleware 2009.

W64 Chris Thompson, Jules White, Brian Dougherty, and Douglas C. Schmidt, "Optimizing Mobile Application Performance with Model-Driven Engineering," Proceedings of the 7th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2009), November 16-18, 2009, Newport Beach, California.

W63 Jules White and Douglas C. Schmidt, "Filtered Cartesian Flattening: An Approximation Technique for Optimally Selecting Features while Adhering to Resource Constraints," proceedings of the Workshop on Analyses of Software Product Lines (ASPL 2008) at the Software Product Lines Conference (SPLC), September 8-12, 2008, Limerick, Ireland.

W62 Joe Hoffert, Douglas C. Schmidt, Mahesh Balakrishnan, and Ken Birman, Supporting Large-scale Continuous Stream Datacenters via Pub/Sub Middleware and Adaptive Transport Protocols, Proceedings of the 2nd workshop on Large-Scale Distributed Systems and Middleware (LADIS 2008), IBM TJ Watson Research Center, Yorktown, New York, September 2008.

W61 Nishanth Shankaran, John S. Kinnebrew, Xenofon D. Koutsoukos, Chenyang Lu, Douglas C. Schmidt, and Gautam Biswas, Towards an Integrated Planning and Adaptive Resource Management Architecture for Distributed Real-time Embedded Systems," Proceedings of the

Workshop on Adaptive and Reconfigurable Embedded Systems (APRES) at the 14th IEEE Real-Time and Embedded Technology and Applications Symposium, St. Louis, MO, United States, April 22 - April 24, 2008.

W60 Serena Fritsch, Aline Senart, Douglas C. Schmidt, and Siobhan Clarke, "Scheduling Time-bounded Dynamic Software Adaptation," Proceedings of the workshop on Software Engineering for Adaptive and Self-Managing Systems at the 30th IEEE/ACM International Conference on Software Engineering May 12-13, 2008, Leipzig, Germany.

W59 James Hill, Jules White, Sean Eade, and Douglas C. Schmidt, "Towards a Solution for Synchronizing Disparate Models of Ultra-Large-Scale Systems," Proceedings of the Second International Workshop on Ultra-Large-Scale Software- Intensive Systems at the 30th IEEE/ACM International Conference on Software Engineering May 10-11, 2008, Leipzig, Germany.

W58 Douglas C. Schmidt and Hans van't Hag, "Addressing the Challenges of Tactical Information Management in Net-Centric Systems with OpenSplice DDS," Proceedings of the 16th International ACM Workshop on Parallel and Distributed Real-Time Systems (WPDRTS '08), Miami, Florida, April 2008.

W57 Shanshan Jiang, Yuan Xue, and Douglas C. Schmidt, "Disruption-Aware Service Composition and Recovery in Dynamic Networking Environments," WoRkshop on Automating Service Quality (WRAQS) 2007, Co-Located with ASE 2007 November 6, 2007, Atlanta, Georgia.

W56 Jules White, Douglas C. Schmidt, Sean Mulligan, "The Generic Eclipse Modeling System," Model-Driven Development Tool Implementer's Forum, TOOLS '07, June, 2007, Zurich, Switzerland.

W55 John S. Kinnebrew, Nishanth Shankaran, Gautam Biswas, and Douglas C. Schmidt, A Decision-Theoretic Planner with Dynamic Component Reconfiguration for Distributed Real-time and Embedded Systems, Proceedings of the Workshop on Artificial Intelligence for Space Applications at IJCAI 2007, January 6-12, 2007, Hyderabad, India.

W54 Andrey Nechypurenko, Jules White, Egon Wuchner, and Douglas C. Schmidt, "Applying Model Intelligence Frameworks for Deployment Problem in Real-time and Embedded Systems," Proceedings of MARTES: Modeling and Analysis of Real-Time and Embedded Systems to be held on October 2, 2006 in Genova, Italy in conjunction with the 9th International Conference on Model Driven Engineering Languages and Systems, MoDELS/UML 2006.

W53 Jules White, Andrey Nechypurenko, Egon Wuchner, and Douglas C. Schmidt, "Intelligence Frameworks for Assisting Modelers in Combinatorically Challenging Domains," Proceedings of the Workshop on Generative Programming and Component Engineering for QoS Provisioning in Distributed Systems, October 23, 2006, Portland, Oregon.

W52 Nishanth Shankaran, Xenofon Koutsoukos, Douglas C. Schmidt, and Aniruddha Gokhale, "Evaluating Adaptive Resource Management for Distributed Real-Time Embedded Systems," Proceedings of the 4th Workshop on Adaptive and Reflective Middleware, November 28, 2005 Grenoble, France.

W51 Jules White and Douglas Schmidt, "Simplifying the Development of Product-line Customization Tools via Model Driven Development," MODELS 2005 workshop on MDD for Software Product-lines: Fact or Fiction?, October 2, 2005, Jamaica.

W50 Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, Venkatesh Prasad Ranganath, and John Hatcliff, "Model-driven Middleware Specialization Techniques for Software Product-line Architectures in Distributed Real-time and Embedded Systems," MODELS 2005 workshop on MDD for Software Product-lines: Fact or Fiction?, October 2, 2005, Jamaica.

W49 Gen Deng, Gunther Lenz, and Douglas C. Schmidt, "Addressing Domain Evolution Challenges in Model-Driven Software Product-line Architectures," MODELS 2005 workshop on MDD for Software Product-lines: Fact or Fiction?, October 2, 2005, Jamaica.

W48 Andrey Nechypurenko and Douglas C. Schmidt, "Supporting Model Reusability with Pattern-based Composition Units," Proceedings of the IST 2nd European Workshop on Model Driven Architecture (MDA), with an emphasis on Methodologies and Transformations September 7th-8th 2004, Canterbury, England.

W47 Cemal Yilmaz, Arvind S. Krishna, Atif Memon, Adam Porter, Douglas C. Schmidt, Aniruddha Gokhale, and Balachandran Natarajan, "A Model-based Distributed Continuous Quality Assurance Process to Enhance the Quality of Service of Evolving Performance-intensive Soft-

ware Systems," Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), Edinburgh, Scotland, UK, May 24, 2004.

W46 Andrey Nechypurenko, Douglas C. Schmidt, Tao Lu, Gan Deng, Aniruddha Gokhale, "Applying MDA and Component Middleware to Large-scale Distributed Systems: a Case Study, Proceedings of the OMG 1st European Workshop on Model Driven Architecture with Emphasis on Industrial Application, Enschede, the Netherlands, March 2004.

W45 Gan Deng, Tao Lu, Emre Turkay, Aniruddha Gokhale, Douglas C. Schmidt, and Andrey Nechypurenko, "Model Driven Development of Inventory Tracking System," Proceedings of the OOPSLA 2003 Workshop on Domain-Specific Modeling Languages, Anaheim, CA, October 2003.

W44 Tao Lu, Emre Turkay, Aniruddha Gokhale, and Douglas C. Schmidt, "CoSMIC: An MDA Tool suite for Application Deployment and Configuration," Proceedings of the OOPSLA 2003 Workshop on Generative Techniques in the Context of Model Driven Architecture, Anaheim, CA, October 2003.

W43 Arvind S. Krishna, Jai Balasubramanian, Aniruddha Gokhale, Douglas C. Schmidt, Diego Sevilla, and Gautham Thaker, "Empirically Evaluating CORBA Component Model Implementations," Proceedings of the ACM OOPSLA 2003 Workshop on Middleware Benchmarking, Anaheim, CA, October 2003.

W42 Aniruddha Gokhale, Douglas C. Schmidt, Tao Lu, Balachandran Natarjan, and Nanbor Wang, CoSMIC: An MDA Generative Tool for Distributed Real-time and Embedded Applications, Workshop on Model-driven Approaches to Middleware Applications Development at 4th IFIP/ACM/USENIX International Conference on Middleware for Distributed Systems Platforms, June 16, 2003, Rio de Janeiro, Brazil.

W41 Ossama Othman, Jaigaesh Balasubramanian, and Douglas C. Schmidt, "The Design and Performance of an Adaptive Middleware Load Balancing and Monitoring Service," Third International Workshop on Self-Adaptive Software, Arlington, VA, USA, June 9-11, 2003.

W40 Radu Cornea, Shivajit Mohapatra, Nikil Dutt, Rajesh Gupta, Ingolf Krueger, Alex Nicolau, Doug Schmidt, Sandeep Shukla, and Nalini Venkatasubramanian, "A Model-Based Approach to System Specification for Distributed Real-time and Embedded Systems," IEEE RTAS Workshop on Model-Driven Embedded Systems, Washington DC, May 27-30, 2003.

W39 Adam Porter, Cemal Yilmaz, and Douglas C. Schmidt "Distributed Continuous Quality Assurance: The Skoll Project," Proceedings of the Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), Portland, Oregon, May 9, 2003.

W38 Krishnakumar Balasubramanian, Douglas C. Schmidt, Nanbor Wang, Christopher D. Gill, "Towards Composable Distributed Real-time and Embedded Software," Proceedings of the $8^{th}$ IEEE Workshop on Object-oriented Real-time Dependable Systems, Guadalajara, Mexico, January 2003.

W37 Aniruddha Gokhale, Balachandran Natarjan, Douglas C. Schmidt, Andrey Nechypurenko, Nanbor Wang, Jeff Gray, Sandeep Neema, Ted Bapty, and Jeff Parsons, "CoSMIC: An MDA Generative Tool for Distributed Real-time and Embedded Component Middleware and Applications," Proceedings of the OOPSLA 2002 Workshop on Generative Techniques in the Context of Model Driven Architecture, Seattle, WA, November 2002.

W36 M. Mousavi, G. Russello, M. Chaudron, M. Reniers, T. Basten, A. Corsaro, S. Shukla, R. Gupta, and D.C. Schmidt, "Using Aspect-GAMMA in Design and Verification of Embedded Systems," Proceedings of the Seventh Annual IEEE International Workshop on High Level Design Validation and Test Workshop, Cannes, France, October 27-29, 2002.

W35 Douglas C. Schmidt, Andy Gokhale, and Chris Gill, "Applying Model-Integrated Computing and DRE Middleware to High Performance Embedded Computing Applications," Proceedings of the 6th Annual Workshop on High-Performance Embedded Computing (HPEC), September 24-26, Boston, MA.

W34 Douglas C. Schmidt, "Adaptive and Reflective Middleware for Distributed Real-time and Embedded Systems," EMSOFT 2002: Second Workshop on Embedded Software, Grenoble, France, October, 7-9th, 2002.

W33 Aniruddha S. Gokhale and Douglas C. Schmidt and Joseph K. Cross and Christopher Andrews and Sylvester J. Fernandez and Bala Natarajan and Nanbor Wang and Chris D. Gill, "Towards

Real-time Support in Fault-tolerant CORBA," IEEE Workshop on Dependable Middleware-Based Systems, Washington, D.C., June 23-26, 2002.

W32 Chris Gill, Joe Loyall, Rick Schantz, and Douglas C. Schmidt, "Lessons Learned From Using Adaptive DOC Middleware in Real Application Contexts," IEEE Workshop on Dependable Middleware-Based Systems, Washington, D.C., June 23-26, 2002.

W31 M. Mousavi, G. Russello, M. Chaudron, M. Reniers, T. Basten, A. Corsaro, S. Shukla, R. Gupta, and D.C. Schmidt, "Aspects + GAMMA = AspectGAMMA A Formal Framework for Aspect-Oriented Specification," Proceedings of Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design Workshop, Enschede, The Netherlands, April 2002.

W30 Joseph K. Cross and Douglas C. Schmidt, "Meta-Programming Techniques for Distributed Real-time and Embedded Systems," Proceedings of the 7th IEEE Workshop on Object-oriented Real-time Dependable Systems, San Diego, CA, January, 2002.

W29 Douglas C. Schmidt and Mayur Deshpande and Carlos O'Ryan, "Operating System Performance in Support of Real-time Middleware," Proceedings of the 7th IEEE Workshop on Object-oriented Real-time Dependable Systems, San Diego, CA, January, 2002.

W28 Christopher D. Gill, Ron Cytron, and Douglas C. Schmidt, "Middleware Scheduling Optimization Techniques for Distributed Real-Time and Embedded Systems," Proceedings of the 7th IEEE Workshop on Object-oriented Real-time Dependable Systems, San Diego, CA, January, 2002.

W27 Douglas C. Schmidt, "Adaptive and Reflective Middleware for Distributed Real-time and Embedded Systems," EMSOFT 2001: First Workshop on Embedded Software, Lake Tahoe, California, October, 8th-10th, 2001.

W26 Darrell Brunsch, Carlos O'Ryan, and Douglas C. Schmidt, "Designing an Efficient and Scalable Server-side Asynchrony Model for CORBA," Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.

W25 Irfan Pyarali, Marina Spivak, Douglas C. Schmidt, and Ron Cytron, "Optimizing Thread-Pool Strategies for Real-Time CORBA," Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.

W24 Yamuna Krishnamurthy, Vishal Kachroo, David A. Karr, Craig Rodrigues, Joseph P. Loyall, Richard Schantz, and Douglas C. Schmidt, "Integration of QoS-enabled Distributed Object Computing Middleware for Developing Next-generation Distributed Applications," Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.

W23 Ossama Othman and Douglas C. Schmidt, Optimizing Distributed system Performance via Adaptive Middleware Load Balancing, Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.

W22 Pradeep Gore, Douglas C. Schmidt, Carlos O'Ryan, and Ron Cytron, "Designing and Optimizing a Scalable CORBA Notification Service," Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.

W21 Douglas C. Schmidt and Adam Porter, "Leveraging Open-Source Processes to Improve the Quality and Performance of Open-Source Software," Proceedings of the 1st Workshop on Open Source Software Engineering, ICSE 23, Toronto, Canada, May 15, 2001.

W20 Christopher D. Gill, David Levine, Douglas C. Schmidt, "Towards Real-Time Adaptive QoS Management in Middleware for Embedded Computing Systems," Fourth Annual Workshop on High Performance Embedded Computing, MIT Lincoln Laboratory, September 20-22, 2000.

W19 Christopher D. Gill, Fred Kuhns, David Levine, Douglas C. Schmidt, Bryan S. Doerr, and Richard E. Schantz, "Applying Adaptive Real-time Middleware to Address Grand Challenges of COTS-based Mission-Critical Real-Time Systems," Proceedings of the 1st International Workshop on Real-Time Mission-Critical Systems: Grand Challenge Problems, IEEE, Phoenix, Arizona, November 30, 1999.

W18 Carlos O'Ryan, Douglas C. Schmidt, David Levine, and Russell Noseworthy, "Applying a Real-time CORBA Event Service to Large-scale Distributed Interactive Simulation," 5th In-

ternational Workshop on Object-oriented Real-Time Dependable Systems, IEEE, Monterey, CA, November 15-18, 1999.

W17 Fred Kuhns, Carlos O'Ryan, Douglas C. Schmidt, and Jeff Parsons, "The Design and Performance of a Pluggable Protocols Framework for Object Request Broker Middleware," Proceedings of the IFIP Sixth International Workshop on Protocols For High-Speed Networks (PfHSN '99), Salem, MA, August 25-27, 1999.

W16 David Levine, Sergio Flores-Gaitan, and Douglas C. Schmidt, "Measuring OS Support for Real-time CORBA ORBs," Proceedings of the Fourth International IEEE Workshop on Object-oriented Real-time Dependable Systems (WORDS'99), Santa Barbara, California, January 27-29, 1999.

W15 Douglas C. Schmidt, Rajeev Bector, David Levine Sumedh Mungee, and Guru Parulkar, "TAO: a Middleware Framework for Real-time ORB Endsystems," Proceedings of the Workshop on Middleware for Real-Time Systems and Services, held in conjunction with IEEE Real-time Systems Symposium, San Francisco, CA, December 2nd, 1997.

W14 Aniruddha Gokhale, Tim Harrison, Douglas C. Schmidt, and Guru Parulkar, "Operating System Support for Real-time CORBA," *Proceedings of the 5th International Workshop on Object-Orientation in Operating Systems*: IWOOOS 1996 workshop, October 27-28, 1996, Seattle, Washington.

W13 Douglas C. Schmidt, Guru Parulkar, and Chuck Cranor, "Gigabit CORBA - High-Performance Distributed Object Computing," Gigabit Networking Workshop (GBN'96), 24 March 1996, San Francisco, in conjunction with INFOCOM '96.

W12 Douglas C. Schmidt, "Acceptor and Connector: Design Patterns for Actively and Passively Initializing Network Services." Workshop on Pattern Languages of Object-Oriented Programs at ECOOP '95, August 7–1, 1995, Aarhus, Denmark.

W11 Douglas C. Schmidt, "High-Performance Event Filtering for Dynamic Multi-point Applications," Proceedings of the 1st Workshop on High Performance Protocol Architectures (HIPPARCH), INRIA, Sophia Antipolis, France, December, 1994, p 1-8.

W10 Douglas C. Schmidt, "Flexible Configuration of High-Performance Object-Oriented Distributed Communication Systems," 9th OOPSLA Conference, invited paper to the Workshop on Flexibility in Systems Software, ACM, Portland, Oregon, October, 1994, pp. 1-4.

W9 Douglas C. Schmidt and Tatsuya Suda, "Measuring the Impact of Alternative Parallel Process Architectures on Communication Subsystem Performance," *Proceedings of the Proceedings of the 4th International Workshop on Protocols for High-Speed Networks*, IFIP, Vancouver, British Columbia, August, 1994, pp. 103-118.

W8 Douglas C. Schmidt and Tatsuya Suda, "The Service Configurator Framework: An Extensible Architecture for Dynamically Configuring Concurrent, Multi-service Network Daemons," *Proceedings of the 2nd International Workshop on Configurable Distributed Systems*, IEEE, Pittsburgh, PA, March 21-23, 1994, pp. 190-201.

W7 Douglas C. Schmidt, Burkhard Stiller, Tatsuya Suda, and Martina Zitterbart, "Tools for Generating Application-Tailored Multimedia Protocols on Heterogeneous Multi-Processor Platforms," *Proceedings of the 2nd Workshop on High-Performance Communications Subsystems*, IEEE, Williamsburg, Virginia, September 1-3, 1993, pp. 1-7.

W6 Douglas C. Schmidt and Tatsuya Suda, "A Framework for Developing and Experimenting with Parallel Process Architectures to Support High-Performance Transport Systems," *Proceedings of the 2nd Workshop on High-Performance Communications Subsystems*, IEEE, Williamsburg, Virginia, September 1-3, 1993, pp. 1-8.

W5 Tatsuya Suda, Douglas C. Schmidt, Donald F. Box, Duke Hong and Hung Huang, "High Speed Networks," *Proceedings of the International Computer World Symposium '92*, Kobe, Japan, November, 1992.

W4 Hung K. Huang, Douglas C. Schmidt, Donald F. Box, Kazu Shimono, Girish Kotmire, Unmesh Rathi, and Tatsuya Suda, "ADAPTIVE: A Prototyping Environment for Transport Systems." *Proceedings of the 4th International Workshop on Computer Aided Modeling, Analysis, and Design of Communication Links and Networks (CAMAD)*, IEEE, Montreal, Canada, September, 1992.

W3 Donald F. Box, Douglas C. Schmidt, and Tatsuya Suda, "Alternative Approaches to ATM/-Internet Interoperation," *Proceedings of the 1st Workshop on the Architecture and Implemen-*

*tation of High-Performance Communication Subsystems*, IEEE, Tucson, Arizona, February 17-19, 1992, pp. 1-5.

W2 Douglas C. Schmidt and Richard Selby "Modeling Software Interconnectivity," *Proceedings of the 22nd Symposium on the Interface: Computer Science and Statistics*, East Lansing, MI, May, 1990.

W1 Richard W. Selby, Greg James, Kent Madsen, Joan Mahoney, Adam A. Porter, and Douglas C. Schmidt "Classification Tree Analysis Using the Amadeus Measurement and Empirical Analysis System," *Proceedings of the 14th Annual Software Engineering Workshop at NASA Software Engineering Laboratory*, College Park, Maryland, November, 1989, pp. 239-250.

- **Trade Magazine and Newsletter Publications**

M74 Douglas C. Schmidt and Jules White, "Why Don't Big Companies Keep Their Computer Systems Up-to-date?," *The Conversation*, September 26, 2017.

M73 Douglas C. Schmidt, Accelerating the Industrial Internet with the OMG Data Distribution Service, Real-time Innovations, January 2014.

M72 Douglas C. Schmidt and Ron Guida, "Elastic Application Platforms for Cloud Computing," HPC In the Cloud, September 2010.

M71 Douglas C. Schmidt and Ron Guida, "Achieving Ultra High Performance in the Cloud," HPC In the Cloud, August 2010.

M70 Douglas C. Schmidt, "Building Ultra High Performance Computing Applications with Zircon Software," Programmer's Paradise, May 31st, 2010.

M69 Egon Wuchner, Andrey Nechypurenko, Jules White, and Douglas C. Schmidt, "Das Generic Eclipse Modeling System (GEMS): Skalierbare Domanenmodellierung Leicht(er) Gemacht," SIGS ObjectSpektrum, June, 2007.

M68 Jules White, Douglas C. Schmidt, Andrey Nechypurenko, and Egon Wuchner, Introduction to the Generic Eclipse Modeling System, Eclipse Magazine, Volume 06, January, 2007.

M67 Bala Natarajan, Douglas C. Schmidt, and Steve Vinoski, "Object Interconnections: The CORBA Component Model Part 4: Implementing Components with CCM, C/C++ Users Journal, October, 2004.

M66 Bala Natarajan, Douglas C. Schmidt, and Steve Vinoski, "The CORBA Component Model Part 3: The CCM Container Architecture and Component Implementation Framework," C/C++ Users Journal, September, 2004.

M65 Douglas C. Schmidt, Richard Schantz, Aniruddha Gokhale, and Joe Loyall, "Middleware R&D Challenges for Distributed Real-time and Embedded Systems," SIGBED Review, Volume 1, No. 1, April 2004.

M64 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: The CORBA Component Model: Part 2: Defining Components with the IDL 3.x Types," C/C++ Users Journal, May, 2004.

M63 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: The CORBA Component Model: Part 1: Evolving Towards Components," C/C++ Users Journal, February, 2004.

M62 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: XML Reflection for CORBA," C/C++ Users Journal, December, 2003.

M61 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: CORBA Metaprogramming Mechanisms, Part 1: Portable Interceptors Concepts and Components," C/C++ Users Journal, March, 2003.

M60 Douglas C. Schmidt and Steve Vinoski, Object Interconnections: "Dynamic CORBA, Part 4: The Interface Repository," C/C++ Users Journal, January, 2003.

M59 Douglas C. Schmidt, "Voice of the Customer: An Interview," Raytheon Technology Today, spring 2003, Volume 2, Issue 1.

M58 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: Dynamic CORBA, Part 3: The Dynamic Skeleton Interface," C/C++ Users Journal, November, 2002.

M57 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: Dynamic CORBA, Part 2: Dynamic Any," C/C++ Users Journal, September, 2002.

M56 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: Dynamic CORBA, Part 1: The Dynamic Invocation Interface," C/C++ Users Journal, July, 2002.

M55 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: Real-time CORBA, Part 4: Protocol Selection and Explicit Binding," C/C++ Users Journal, May, 2002.

M54 Douglas C. Schmidt and Steve Huston, "Why Standards Alone Won't Get You Portable Software And How to Make Open Source Development Work for You," InformIT: Focus on C++, Addison-Wesley.

M53 Douglas C. Schmidt and Steve Vinoski, "Real-time CORBA Part 3: Thread Pools and Synchronizers," C/C++ Users Journal, March, 2002.

M52 Douglas C. Schmidt and Steve Vinoski, "Real-time CORBA, Part 2: Applications and Priorities," C/C++ Users Journal, January, 2002.

M51 Douglas C. Schmidt and Steve Vinoski, "Real-time CORBA, Part 1: Motivation and Overview," *C/C++ Users Journal*, October, 2001.

M50 Douglas C. Schmidt and Steve Vinoski, "CORBA and XML, Part 3: SOAP and Web Services," *C/C++ Users Journal*, September, 2001.

M48 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: CORBA and XML, Part 2: XML as CORBA Data," *C/C++ Users Journal*, July, 2001.

M47 Douglas C. Schmidt and Steve Vinoski, "Object Interconnections: CORBA and XML, Part 1: Versioning," *C/C++ Users Journal*, May, 2001.

M46 Douglas C. Schmidt and Steve Vinoski, "Standard C++ and the OMG C++ Mapping: Server-side Mappings and Pseudo-Objects," *C/C++ Users Journal*, April, 2001.

M45 Douglas C. Schmidt and Steve Vinoski, "Standard C++ and the OMG C++ Mapping," *C/C++ Users Journal*, January, 2001.

M44 Douglas C. Schmidt and Steve Vinoski, "The History of the OMG C++ Mapping", *C/C++ Users Journal*, November, 2000.

M43 Douglas C. Schmidt and Steve Vinoski, "An Overview of the OMG CORBA Messaging Quality of Service (QoS) Framework," *C++ Report*, SIGS, Vol. 12, No 3, March, 2000.

M42 Douglas C. Schmidt, "Monitor Object – an Object Behavior Pattern for Concurrent Programming," *C++ Report*, SIGS, Vol. 12., No. 4. May 2000.

M41 Alexander B. Arulanthu, Carlos O'Ryan, Douglas C. Schmidt, and Michael Kircher, "Applying Patterns and Components to Develop an IDL Compiler for CORBA AMI Callbacks," *C++ Report*, SIGS, Vol. 12, No. 3, March, 2000.

M40 David Levine, Chris Gill, and Douglas C. Schmidt, "Object Lifetime Manager – A Complementary Pattern for Controlling Object Creation and Destruction," *C++ Report*, SIGS, Vol. 11, No. 11, November/December, 1999.

M39 Douglas C. Schmidt, Steve Vinoski, and Nanbor Wang, "Collocation Optimizations for CORBA," *C++ Report*, SIGS, Vol. 11, No. 10, October, 1999.

M38 Douglas C. Schmidt, "Strategized Locking, Thread-safe Decorator, and Scoped Locking: Patterns and Idioms for Simplifying Multi-threaded C++ Components," *C++ Report*, SIGS, Vol. 11, No. 9, September, 1999.

M37 Douglas C. Schmidt and Steve Vinoski, "Time-Independent Invocation and Interoperable Routing," *C++ Report*, SIGS, Vol. 11, No 5, May, 1999.

M36 Michael Kircher and Douglas C. Schmidt, "Dove: A Distributed Object Visualization Environment," *C++ Report*, SIGS, Vol. 11, No 3, March, 1999.

M35 Douglas C. Schmidt, "Wrapper Facade: A Structural Pattern for Encapsulating Functions within Classes," *C++ Report*, SIGS, Vol. 11, No 2, February, 1999.

M34 Douglas C. Schmidt and Steve Vinoski, "Programming Asynchronous Method Invocation with CORBA Messaging," *C++ Report*, SIGS, Vol. 11, No 2, February, 1999.

M33 Douglas C. Schmidt "Why Software Reuse has Failed and How to Make It Work for You," C++ Report, SIGS, Vol. 11, No. 1, January, 1999.

M32 Douglas C. Schmidt, "An Architectural Overview of the ACE Framework: A Case-study of Successful Cross-platform Systems Software Reuse," USENIX login magazine, Tools special issue, November, 1998.

M31 Douglas C. Schmidt, "GPERF: A Perfect Hash Function Generator," C++ Report, SIGS, Vol. 10, No. 10, November/December, 1998.

M30 Douglas C. Schmidt and Steve Vinoski, "Introduction to CORBA Messaging," SIGS, Vol. 10, No 10, November/December, 1998.

M29 Douglas C. Schmidt and Steve Vinoski, "C++ Servant Managers for the Portable Object Adapter," SIGS, Vol. 10, No 8, September, 1998.

M28 Chris Cleeland and Douglas C. Schmidt, "External Polymorphism, An Object Structural Pattern for Transparently Extending C++ Concrete Data Types," C++ Report, SIGS, Vol. 10, No. 6, July/August, 1998.

M27 Douglas C. Schmidt and Irfan Pyarali, "Strategies for Implementing POSIX Condition Variables on Win32," C++ Report, SIGS, Vol. 10, No. 5, June, 1998.

M26 Douglas C. Schmidt and Steve Vinoski, "Using the Portable Object Adapter for Transient and Persistent CORBA Objects," C++ Report, SIGS, Vol. 10, No 4. May, 1998.

M25 Douglas C. Schmidt, "Applying Design Patterns to Simplify Signal Handling," C++ Report, SIGS, Vol. 10, No. 4, May, 1998.

M24 Douglas C. Schmidt, Tim H. Harrison, and Nat Pryce, "Thread-specific Storage: an Object Behavioral Pattern for Efficiently Accessing per-Thread State," C++ Report, SIGS, Vol. 9, No. 10, November/December, 1997

M23 Douglas C. Schmidt and Steve Vinoski, "Object Adapters: Concepts and Terminology," C++ Report, SIGS, Vol. 9, No 11. November/December, 1997.

M22 Prashant Jain and Douglas C. Schmidt, "Dynamically Configuring Communication Services with the Service Configurator Pattern," *C++ Report*, SIGS, Vol. 9, No. 6, June, 1997.

M21 Douglas C. Schmidt and Steve Vinoski, "Overcoming Drawbacks in the OMG Events Service," *C++ Report*, SIGS, Vol. 9, No 6. June, 1997.

M20 Douglas C. Schmidt and Steve Vinoski, "OMG Event Object Service," *C++ Report*, SIGS, Vol. 9, No 2. February, 1997.

M19 Prashant Jain and Douglas C. Schmidt, "Experiences Converting a C++ Communication Framework to Java," *C++ Report*, SIGS, Vol. 9, No. 1, January, 1997.

M18 Douglas C. Schmidt, "Lessons Learned Building Reusable OO Telecommunication Software," Multiuse Express, Lucent Technologies, Vol. 4, No. 6, December, 1996.

M17 Douglas C. Schmidt and Steve Vinoski, "Distributed Callbacks and Decoupled Communication in CORBA," *C++ Report*, SIGS, Vol. 8, No 9. October, 1996.

M16 Timothy H. Harrison and Douglas C. Schmidt, "Evaluating the Performance of OO Network Programming Toolkits," *C++ Report*, SIGS, Vol. 8, No 7. July/August 1996.

M15 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Programming Techniques for Multi-threaded Servers – the Thread-per-Session Concurrency Model," *C++ Report*, SIGS, Vol. 8, No 7. July/August 1996.

M14 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Programming Techniques for Multi-threaded Servers – the Thread-Pool Concurrency Model," *C++ Report*, SIGS, Vol. 8, No 4. April 1996.

M13 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Programming Techniques for Multi-threaded Servers – the Thread-per-Request Concurrency Model," *C++ Report*, SIGS, Vol. 8, No 2. February 1996.

M12 Douglas C. Schmidt, "A Design Pattern for Actively Initializing Network Services," *C++ Report*, SIGS, Vol. 8, No. 1, January 1996.

M11 Douglas C. Schmidt, "Design Patterns for Initializing Network Services: Introducing the Acceptor and Connector Patterns," *C++ Report*, SIGS, Vol. 7, No. 9, November/December 1995.

M10 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Server-side Distributed Programming Techniques," Object Interconnections Column, *C++ Report*, SIGS, Vol. 7, No. 8, October 1995.

M9 Douglas C. Schmidt and Steve Vinoski, "Comparing Alternative Client-side Distributed Programming Techniques," Object Interconnections Column, *C++ Report*, SIGS, Vol. 7, No. 4, May 1995.

M8 Douglas C. Schmidt and Paul Stephenson, "Using Design Patterns to Evolve System Software from UNIX to Windows NT," *C++ Report*, SIGS, Vol. 7, No. 3, March/April 1995, pp. 27-39.

M8 Douglas C. Schmidt and Steve Vinoski, "Distributed Object Computing by Example," Object Interconnections Column, *C++ Report*, SIGS, Vol. 7, No. 2, February 1995.

M7 Douglas C. Schmidt and Steve Vinoski, "Distributed Object Computing with C++," Object Interconnections Column, *C++ Report*, SIGS, Vol. 7, No. 1, January 1995.

M6 Douglas C. Schmidt, "Transparently Parameterizing Synchronization Mechanisms into a Concurrent Distributed Application," *C++ Report*, SIGS, Vol. 6, No. 5, July/August 1994, pp. 1-10.

M5 Douglas C. Schmidt, "A Domain Analysis of Network Daemon Design Dimensions," *C++ Report*, SIGS, Vol. 6, No. 3, March/April, 1994, pp. 1-12.

M4 Douglas C. Schmidt, "The Object-Oriented Design and Implementation of the Reactor: A C++ Wrapper for UNIX I/O Multiplexing," *C++ Report*, SIGS, Vol. 5, No. 7, September, 1993, pp. 1-14.

M3 Douglas C. Schmidt, "The Reactor: An Object-Oriented Interface for Event-Driven UNIX I/O Multiplexing," *C++ Report*, SIGS, Vol. 5, No. 2, February, 1993, pp. 1-12.

M2 Douglas C. Schmidt, "IPC_SAP: An Object-Oriented Interface to Operating System Interprocess Communication Services," *C++ Report*, SIGS, Vol. 4, No. 8, November/December, 1992, pp. 1-10.

M1 Douglas C. Schmidt, "Systems Programming with C++ Wrappers: Encapsulating Interprocess Communication Services with Object-Oriented Interfaces," *C++ Report*, SIGS, Vol. 4, No. 7, September/October, 1992, pp 1-6.

- **Refereed Short Papers, Posters, and Demos**

P14 Maria Powell, Marcelino Rodriguez Cancio, David Young, William Nock, Beshoy Abdelmessih, Amy Zeller, Irvin Perez Morales, Peng Zhang, C. Gaelyn Garrett, Douglas Schmidt, Jules White, and Alexander Gelbard, "Decoding Phonation with Artifical Intelligence: Proof of Concept," Poster Proceedings in the 13th International Conference on Advances in Quantitative Laryngology, Voice and Speech Research, Montreal, Canada, June 2-4th, 2019.

P13 Akram Hakiri, Berthou Pascal, Gayraud Thierry, Aniruddha Gokhale, Joe Hoffert, and Douglas C. Schmidt, "SIP-based QoS Support and Session Management for DDS-based Distributed Real-time and Embedded Systems," Poster Proceedings of the 5th ACM International Conference on Distributed Event-based Systems (DEBS'11), New York City, NY, USA, July 11-15, 2011.

P12 Jules White, Andrey Nechypurenko, Egon Wuchner, and Douglas C. Schmidt "Automatic Role-based Constraint Solving for Real-Time and Embedded Systems: An Approach to Modeling Guidance", poster paper at the 14th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), March 26th-29th, 2007, Tucson, Arizona.

P11 John Kinnebrew, Nishanth Shankaran, Gautam Biswas, and Douglas Schmidt, "A Decision-Theoretic Planner with Dynamic Component Reconfiguration for Distributed Real-Time Applications," Poster paper at the Twenty-First National Conference on Artificial Intelligence, July 16 20, 2006, Boston, Massachusetts.

P10 Jai Balasubramanian, Nishanth Shankar, Douglas C. Schmidt, Gautam Biswas, Patrick Lardieri, Ed Mulholland, and Tom Damiano, "A Framework for (Re)Deploying Components in Distributed Realtime and Embedded Systems," poster paper at the Dependable and Adaptive Distributed Systems, Track of the 21st ACM Symposium on Applied Computing, April 23 -27, 2006, Bourgogne University, Dijon, France.

P9 Jules White and Douglas C. Schmidt, Reducing Enterprise Product Line Architecture Deployment Costs via Model-Driven Deployment and Configuration Testing, Poster paper at the 13th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '06) March 27th-30th, 2006, University of Potsdam, Potsdam, Germany.

P8 Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, and Venkatesh Prasad Ranganat, "Towards Highly Optimized Real-time Middleware for Software Product-line Architectures," Proceedings of the Work-In-Progress session at the 26th IEEE Real-Time Systems Symposium, December 5-8, 2005, Miami, Florida.

P7 Gan Deng, Douglas Schmidt, and Aniruddha Gokhale, "Supporting Configuration and Deployment of Component-based DRE Systems Using Frameworks, Models, and Aspects," Poster

Session of the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programing, Systems, Languages and Applications (OOPSLA 2005), San Diego, CA, October 16-20, 2005.

P6 Jules White, Douglas Schmidt, and Aniruddha Gokhale, "The J3 Process for Building Autonomic Enterprise Java Bean Systems," Proceedings of the International Conference on Autonomic Computing (ICAC 2005), Seattle, WA, June 2005 (short paper).

P5 Aniruddha Gokhale, Arvind Krishna and Douglas C. Schmidt, "CoSMIC: Addressing Crosscutting Deployment and Configuration Concerns of Distributed Real-time and Embedded Systems via Aspect-oriented and Model-driven Software Development," Demo Session, Fourth International Conference on Aspect-oriented Software Development (AOSD), Chicago, IL, Mar 2005.

P4 Aniruddha Gokhale, Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Arvind Krishna, George Edwards, Gan Deng, Jeff Parsons, Tao Lu, Balachandran Natarajan and Douglas C. Schmidt, "CoSMIC: Addressing Crosscutting Deployment and Configuration Concerns in QoS-sensitive Distributed Systems," 19th ACM OOPSLA Conference, Poster Session, Vancouver, Canada, Oct 2004.

P3 Joseph K. Cross and Douglas C. Schmidt, "Meta-Programming Techniques to Declaratively Optimize Middleware Policies and Mechanisms," Poster session at the IFIP/ACM Middleware 2001 International Conference on Distributed Systems Platforms, Heidelberg, Germany, November 12-16, 2001.

P2 Aniruddha Gokhale and Douglas C. Schmidt, "Design Principles and Optimizations for High Performance ORBs," ACM, *OOPSLA 97*, Poster Session, Oct 1997, Atlanta, GA, USA.

P1 Douglas C. Schmidt, "Performance Experiments on Alternative Methods for Structuring Active Objects in High-Performance Parallel Communication Systems," 9th ACM OOPSLA Conference, poster paper, Portland, Oregon, October, 1994, pp. 1-12.

- **Editorials and Book Forewords**

E101 Douglas C. Schmidt, "Top 10 Blog Posts of 2020," SEI Blog, January 11th, 2021.

E100 Tom Evans, Mike Gagliardi, Joe Kostial, Nicholas Reimer, and Douglas C. Schmidt, "Shifting from Software Sustainment to Software Engineering in the DoD," SEI Blog, November 30th, 2020.

E99 Douglas C. Schmidt, "The Latest Work from the SEI: Microservices, Ransomware, and Agile in Government," SEI Blog, September 7th, 2020.

E98 Douglas C. Schmidt, "The Latest Work from the SEI: DevSecOps, Artificial Intelligence, and Cybersecurity Maturity Model Certification," SEI Blog, May 25th, 2020.

E97 Douglas C. Schmidt, "The Latest Work from the SEI: Penetration Testing, Artificial Intelligence, and Incident Management," SEI Blog, January 20th, 2020.

E96 Douglas C. Schmidt, "Top 10 Blog Posts of 2019," SEI Blog, January 6th, 2020.

E95 Douglas C. Schmidt, "The Latest Work from the SEI: Microservices, Ransomware, and Agile in Government", SEI Blog, September 7th, 2019.

E94 Douglas C. Schmidt, "The Latest Work from the SEI: DevSecOps, Artificial Intelligence, and Cybersecurity Maturity Model Certification", SEI Blog, May 25th, 2020.

E93 Douglas C. Schmidt, "The Latest Work from the SEI: Penetration Testing, Artificial Intelligence, and Incident Management", SEI Blog, January 20th, 2019.

E92 Douglas C. Schmidt, "The Top 10 Blog Posts of 2019, SEI Blog, January 6th, 2020.

E91 Nick Guertin, Douglas C. Schmidt, and William Scherlis, "Impacts and Recommendations for Achieving Modular Open Systems Architectures, SEI Blog, September 23rd, 2019.

E90 Douglas C. Schmidt, "The Latest Work from the SEI: AI, Deepfakes, Automated Alert Handling, and Cyber Intelligence," SEI Blog, September 2nd, 2019.

E89 Douglas C. Schmidt, "The Latest Research from the SEI in DevSecOps, Threat Modeling, and Insider Threat," SEI Blog, May 28th, 2019.

E88 Douglas C. Schmidt, "Deep Learning, Agile-DevOps, and Cloud Security: The Top 10 Blog Posts of 2018," SEI Blog, January 7th, 2019.

E87 Nick Guertin, Douglas C. Schmidt, and William Scherlis, "The Technical Architecture for Product Line Acquisition in the DoD," SEI Blog, May 6th, 2019.

E86 Nick Guertin, Douglas C. Schmidt, and William Scherlis, "The Organizational Impact of a Modular Product Line Architecture in DoD Acquisition," SEI Blog, April 29th, 2019.

E85 Nick Guertin, Douglas C. Schmidt, and William Scherlis, "Towards a New Model of Acquisition: Product-Line Architectures for the DoD," SEI Blog, March 11th, 2019.

E84 Douglas C. Schmidt, "Deep Learning, Agile-DevOps, and Cloud Security: The Top 10 Blog Posts of 2018," SEI Blog, January 7th, 2019.

E83 Nick Guertin, Douglas C. Schmidt, and William Scherlis, "Emerging Opportunities in Modularity and Open Systems Architectures," SEI Blog, October 15th, 2018.

E82 Douglas C. Schmidt, "Learning, Cyber Intelligence, Managing Privacy and Security, and Network Traffic Analysis: The Latest Work from the SEI," SEI Blog, July 2, 2018.

E81 Douglas C. Schmidt, "Virtual Integration, Blockchain Programming, and Agile/DevOps: The Latest Work from the SEI," SEI Blog, May 28, 2018.

E80 Douglas C. Schmidt, "Fighting Chance: Arming the Analyst in the Age of Big Data," SEI Blog, March 26, 2018.

E79 Douglas C. Schmidt, "Agile/DevOps, Best Practices in Insider Threat, and Dynamic Design Analysis: The Latest Work from the SEI," SEI Blog, February 26, 2018.

E78 Douglas C. Schmidt, "Bitcoin, Blockchain, Machine Learning, and Ransomware: The Top 10 Posts of 2017," SEI Blog, January 8, 2018.

E77 Douglas C. Schmidt, "Cyber Warfare, Technical Debt, Network Border Protection, and Insider Threat: The Latest Work from the SEI," SEI Blog, November 27, 2017.

E76 Douglas C. Schmidt, "Coordinated Vulnerability Disclosure, Ransomware, Scaling Agile, and Android App Analysis: The Latest Work from the SEI,SEI Blog, " September 5, 2017.

E75 Douglas C. Schmidt, "Top 10 SEI Blog Posts of 201I," SEI Blog, July 10, 2017.

E74 Douglas C. Schmidt, "Supply Chain Risk Management, Network Situational Awareness, Software Architecture, and Network Time Protocol: The Latest Work from the SEI," SEI Blog, July 3, 2017.

E73 Douglas C. Schmidt, "Software Assurance, Data Governance, and Malware Analysis: The Latest Work from the SEI," SEI Blog, April 10, 2017.

E72 Douglas C. Schmidt, "Preventing DDoS Attacks, Scaling Agile, Insider Threat, and Software Architecture: The Latest Work from the SEI," SEI Blog, January 30th, 2017.

E71 Douglas C. Schmidt, "Autonomy, Robotics, Verification, DDoS Attacks, and Software Testing: The Top 10 Posts of 2016," SEI Blog, December 19th, 2016.

E70 Douglas C. Schmidt, "Cybersecurity Engineering, Performance, Risk, and Secure Coding: The Latest Work from the SEI," SEI Blog, November 28th, 2016.

E69 Douglas C. Schmidt, "Resilience, Secure Coding, Data Science, Insider Threat, and Scheduling: The Latest Research from the SEI," SEI Blog, October 17th, 2016.

E68 Douglas C. Schmidt, "Data Science, Blacklists, and Mixed-Critical Software: The Latest Research from the SEI," SEI Blog, September 5th, 2016.

E67 Douglas C. Schmidt and Carol Sledge, "A Naval Perspective on Open Systems Architecture," SEI Blog, July 11th, 2016.

E66 Douglas C. Schmidt, "Top 10 SEI Blog Posts of 2016," SEI Blog, July 4th, 2016.

E65 Douglas C. Schmidt, "Top 10 SEI Blog Posts of 2016, SEI Blog, July 4th, 2016. Douglas C. Schmidt, Situational Analysis, Software Architecture, Insider Threat, Threat Modeling, and Honeynets: The Latest Research from the SEI," SEI Blog, May 30th, 2016.

E64 Douglas C. Schmidt, "Threat Analysis Mapping, Connected Vehicles, Emerging Technologies, and Cyber-Foraging: The Latest Research from the SEI," SEI Blog, May 2nd, 2016.

E63 Douglas C. Schmidt, "The Top 10 Blog Posts of 2015: Technical Debt, DevOps, Graph Analytics, Secure Coding, and Testing," January 4th, 2016.

E62 Carol Sledge and Douglas C. Schmidt, "A Discussion on Open-Systems Architecture," SEI Blog November 23rd, 2015.

E61 Douglas C. Schmidt, "Agile, Architecture Fault Analysis, the BIS Wassenaar Rule, and Computer Network Design," SEI Blog, September 7, 2015.

E60 Douglas C. Schmidt, "Testing, Agile Metrics, Fuzzy Hashing, Android, and Big Data" SEI Blog, July 13, 2015.

E59 Douglas C. Schmidt, "Resilience, Model-Driven Engineering, Software Quality, and Android App Analysis," SEI Blog, May 18, 2015.

E58 Douglas C. Schmidt, "Resilience, Metrics, Sustainment, and Software Assurance," SEI Blog, February 23, 2015.

E57 Douglas C. Schmidt, "Software Assurance, Social Networking Tools, Insider Threat, and Risk Analysis," SEI Blog, January 19th 2015.

E56 Douglas C. Schmidt, "The 2014 Year in Review," SEI blog, December 22nd, 2014.

E55 Douglas C. Schmidt, Android, Heartbleed, Testing, and DevOps: An SEI Blog Mid-Year Review, SEI blog, June 30th, 2014.

E54 Douglas C. Schmidt, "The Importance of Automated Testing in Open Systems Architecture Initiatives," SEI blog, March 23rd, 2014.

E53 Douglas C. Schmidt, "How Vanderbilt's Secret Software Lab Is Saving America," gizmodo.com, January 10th, 2014.

E52 Douglas C. Schmidt, "The SEI Blog: the Research Year in Review," SEI blog, December 23rd, 2013.

E51 Douglas C. Schmidt "The Architectural Evolution of DoD Combat Systems," SEI blog, November 25th, 2013.

E50 Douglas C. Schmidt, "Three Qs: Vanderbilt Professor Douglas Schmidt," GE's Industrial Internet blog, September 10th, 2013.

E49 Douglas C. Schmidt, "Towards Affordable DoD Combat Systems in the Age of Sequestration," SEI Blog, September 9th, 2013.

E48 Douglas C. Schmidt, "Ten Tech Terms Everyone Needs to Know for 2014," Yahoo Tech News, August 12, 2013.

E47 Douglas C. Schmidt and Philippe Fauchet, "Students Must Stay to Better Workforce," The Tennessean, August 6th, 2013.

E46 Douglas C. Schmidt, "Learning in MOOC Years," Vanderbilt Magazine, Spring 2013.

E45 Douglas C. Schmidt, The SEI Blog: A Two-year Retrospective, SEI blog, April 1st, 2013.

E44 Douglas C. Schmidt, 2012: "The Research Year in Review," SEI blog, December 24th, 2012.

E43 Douglas C. Schmidt, "Reflections on 20 Years of Architecture for Distributed Real-time and Embedded Systems by Douglas C. Schmidt," SEI blog, October 29th, 2012.

E42 Douglas C. Schmidt, "Applying Agility to DoD Common Operating Platform Environment Initiatives," SEI blog, July 30th, 2012.

E41 Douglas C. Schmidt, "Balancing Agility and Discipline at Scale," SEI blog, July 23rd, 2012.

E40 Douglas C. Schmidt, "Strategic Management of Architectural Technical Debt," SEI blog, July 16th, 2012.

E39 Douglas C. Schmidt, "Agile Methods: Tools, Techniques, and Practices for the DoD Community," SEI blog, July 9th, 2012.

E38 Douglas C. Schmidt, "Applying Agile at-Scale for Mission-Critical Software-Reliant Systems," SEI blog, July 2nd, 2012.

E37 Douglas C. Schmidt, "Toward Common Operating Platform Environments, Part 2: Understanding Success Drivers," SEI blog, May 7th, 2012.

E36 Douglas C. Schmidt, "Toward Common Operating Platform Environments, Part 1: Doing More for Less," SEI blog, April 30th, 2012.

E35 Douglas C. Schmidt, "The Road Ahead for SEI R&D in 2012," SEI blog, December 26th, 2011.

E34 Douglas C. Schmidt, "A Summary of Key SEI R&D Accomplishments in 2011," SEI blog, December 19th, 2011.

E33 Douglas C. Schmidt, "Bridging the Valley of Disappointment for DoD Software Research with SPRUCE," SEI blog, November 7th, 2011.

E32 Douglas C. Schmidt, "The Growing Importance of Software Sustainment for the DoD, Part 2: SEI R&D Activities Related to Sustaining Software for the DoD", SEI blog, August 15th, 2011.

E31 Douglas C. Schmidt, "The Growing Importance of Software Sustainment for the DoD, Part 1: Current Trends and Challenges", SEI blog, August 1st, 2011.

E30 Douglas C. Schmidt, "New and Upcoming SEI Research Initiatives," SEI blog, February 21st, 2011.

E29 Douglas C. Schmidt, "Advancing the Scope and Impact of SEI Research," SEI blog, February 9th, 2011.

E28 Douglas C. Schmidt, Foreword to the book *Patterns xof Parallel Software Design*, by Jorge Luis Ortega Arjona, Wiley, 2010.

E27 Douglas C. Schmidt, Foreword to the book *Practical Software Factories in .NET* by Gunther Lenz and Christoph Wienands, Apress, 2006.

E26 Douglas C. Schmidt, Guest editorial of the IEEE Computer Special Issue on Model Driven Development, February 2006.

E25 Douglas C. Schmidt, Guest editorial for IEEE Networks magazine Special Issue on Middleware Technologies for Future Communication Networks, January 2004.

E24 Douglas C. Schmidt, Foreword to the book *Fundamentals of Distributed Object Systems: The CORBA Perspective*, by Zahir Tari and Omran Bukhres, Wiley and Sons, 2001.

E23 Douglas C. Schmidt, Foreword to the book *Design Patterns in Communication Software*, edited by Linda Rising and published by Cambridge University Press, 2000.

E22 Douglas C. Schmidt, "Trends in Distributed Object Computing" editorial for the special issue on Distributed Object-Oriented Systems appearing in the Parallel and Distributed Computing Practices journal, edited by Maria Cobb and Kevine Shaw, Vol. 3, No. 1, March 2000.

E21 Douglas C. Schmidt, "Object-Oriented Application Frameworks," guest editorial for the of the Communications of the ACM, Special Issue on Object-Oriented Application Frameworks, Vol. 40, No. 10, October 1997.

E20 Douglas C. Schmidt, "Recent Advances in Distributed Object Computing," guest editorial for the IEEE Communications Magazine feature topic issue on Distributed Object Computing, Vol. 14, No. 2, February, 1997.

E19 Douglas C. Schmidt, Guest editorial for the USENIX Computing Systems Special Issue on Distributed Object Computing Vol. 9, No. 4, November/December, 1996.

E18 Douglas C. Schmidt, "Software Patterns," guest editorial for Communications of the ACM, Special Issue on Patterns and Pattern Languages, Vol. 39, No. 10, October 1996.

E17 Douglas C. Schmidt, "Using Design Patterns to Develop Reuseable Object-Oriented Software," Strategic Directions in Computing Research OO Working Group conference, MIT, June 14-15, 1996.

E16 Douglas C. Schmidt, "The Last Waltz," C++ Report, SIGS, Vol. 11, No. 4, April 1999.

E15 Douglas C. Schmidt, "Patterns++ - the Next Generation," C++ Report, SIGS, Vol. 9, No. 4, April 1997.

E14 Douglas C. Schmidt, "CORBA: CASE for the late '90s?" C++ Report, SIGS, Vol. 9, No. 2, February 1997.

E13 Douglas C. Schmidt, "Java: Friend or Foe," C++ Report, SIGS, Vol. 9, No. 1, January 1997.

E12 Douglas C. Schmidt, "Promise Keepers," C++ Report, SIGS, Vol. 8, No. 11, November/December 1996.

E11 Douglas C. Schmidt, "The Programming Principle," C++ Report, SIGS, Vol. 8, No. 10, October 1996.

E10 Douglas C. Schmidt, "Pattern Forces," C++ Report, SIGS, Vol. 8, No. 9, September 1996.

E9 Douglas C. Schmidt, "The Secrets of Success for C++," C++ Report, SIGS, Vol. 8, No. 9, August 1996.

E8 Douglas C. Schmidt, "The C++ Decade," C++ Report, SIGS, Vol. 8, No. 9, August 1996.

E7 Douglas C. Schmidt, "Addressing the Challenge of Concurrent and Distributed Systems," C++ Report, SIGS, Vol. 8, No. 7, July 1996.

E6 Douglas C. Schmidt, "Delivering the Goods," C++ Report, SIGS, Vol. 8, No. 6, June 1996.

E5 Douglas C. Schmidt, "Problems with Process," C++ Report, SIGS, Vol. 8, No. 5, May 1996.

E4 Douglas C. Schmidt, "The Impact of Social Forces on Software Project Failures," C++ Report, SIGS, Vol. 8, No. 4, April 1996.

E3 Douglas C. Schmidt, "Reality Check," C++ Report, SIGS, Vol. 8, No. 3, March 1996.

E2 Douglas C. Schmidt, "Role Models for Success," C++ Report, SIGS, Vol. 8, No. 2, February 1996.

E1 Douglas C. Schmidt, "A Zest for Programming," C++ Report, SIGS, Vol. 8, No. 1, January 1996.

- **Technical Reports**

TR18 Douglas C. Schmidt, "Google Data Collection," Vanderbilt University Technical Report #ISIS-20-201, August 15, 2018.

TR17 Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale, " Ensuring Deployment Predictability of Distributed Real-time and Embedded Systems," Vanderbilt University Technical Report #ISIS-07-814, November 2007.

TR16 Jaiganesh Balasubramanian, Sumant Tambe, Chenyang Lu, Christopher Gill, Aniruddha Gokhale, and Douglas C. Schmidt, "FLARe: a Fault-tolerant Lightweight Adaptive Real-time Middleware for Distributed Real-time and Embedded, Systems," Vanderbilt University Technical Report #ISIS-07-812, October 2007.

TR15 Shanshan Jiang, Yuan Xue, and Douglas Schmidt, "Minimum Disruption Service Composition and Recovery in Mobile Ad hoc Networks," Vanderbilt University Technical Report #ISIS-06-711, December 2006.

TR14 Andrey Nechypurenko, Egon Wuchner, Jules White, Douglas C. Schmidt, " Application of Aspect-based Modeling and Weaving for Complexity Reduction in the Development of Automotive Distributed Real-time Embedded Systems," Vanderbilt University Technical Report #ISIS-06-709, July 2006.

TR13 James H. Hill, John M. Slaby, Steve Baker, Douglas C. Schmidt, "Predicting the Behavior for Components of the SLICE Scenario," Vanderbilt University Technical Report #ISIS-05-608, October 2005.

TR12 Stoyan Paunov, James Hill, Douglas C. Schmidt, John Slaby,and Steve Baker, "Domain-Specific Modeling Languages for Configuring and Evaluating Enterprise DRE System Quality of Service, Vanderbilt University Technical Report #ISIS-05-606, August 2005.

TR11 John M. Slaby, Steve Baker, James Hill, Doug Schmidt, "Applying System Execution Modeling Tools to Evaluate Enterprise Distributed Real-time and Embedded System QoS," Vanderbilt University Technical Report #ISIS-05-604, June 2005.

TR10 Fred Kuhns and Carlos O'Ryan and Douglas C. Schmidt and Jeff Parsons, "The Design and Performance of a Pluggable Protocols Framework for Object Request Broker Middleware," Washington University Technical Report #WUCS-99-12, St. Louis, MO, Dept. of Computer Science, April 1999.

TR9 Sumedh Mungee, Nagarajan Surendran, and Douglas C. Schmidt, "The Design and Performance of a CORBA Audio/Video Streaming Service," Washington University Technical Report #WUCS-98-15.

TR8 Lutz Prechelt, Barbara Unger, Douglas C. Schmidt, "Replication of the First Controlled Experiment on the Usefulness of Design Patterns: Detailed Description and Evaluation." 77 pgs., Washington University Technical Report #wucs-97-34, December 1997.

TR7 Aniruddha Gokhale and Douglas C. Schmidt, "Optimizing the Performance of the CORBA Internet Inter-ORB Protocol Over ATM," Washington University Technical Report #WUCS-97-10.

TR6 James Hu and Sumedh Mungee and Douglas C. Schmidt, "Principles for Developing and Measuring High-performance Web Servers over ATM," Washington University Technical Report #WUCS-97-10.

TR5 Chris Cleeland, Douglas C. Schmidt, and Tim H. Harrison, "External Polymorphism – An Object Structural Pattern for Transparently Extending Concrete Data Types," The 3rd annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 4-6, 1996, Washington University Technical Report #WUCS-97-07.

TR4 Timothy H. Harrison, Douglas C. Schmidt, and Irfan Pyarali, "Asynchronous Completion Token," The 3rd annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 4-6, 1996, Washington University Technical Report #WUCS-97-07.

TR3 Douglas C. Schmidt and Timothy H. Harrison, "The Double-Checked Locking Pattern," The 3rd annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 4-6, 1996, Washington University Technical Report #WUCS-97-07.

TR2 Prashant Jain and Douglas C. Schmidt, "The Service Configurator Pattern," The 3rd annual Pattern Languages of Programming conference in Allerton Park, Illinois, September 4-6, 1996, Washington University Technical Report #WUCS-97-07.

TR1 Douglas C. Schmidt, "Acceptor and Connector: Design Patterns for Initializing Network Services," The EuroPLoP '96 conference in Kloster Irsee, Germany, July 10-14, 1996, Washington University Technical Report #WUCS-97-07.

# Presentations

### Conference Presentations

1. "Mobile Applications Technology Overview," Digital Technologies in Cancer Research Workshop, Vanderbilt University, Nashville, TN, May 15th 2019.
2. "Website Applications Technology Overview," Digital Technologies in Cancer Research Workshop, Vanderbilt University, Nashville, TN, May 15th 2019.
3. "Producing and Delivering a Coursera MOOC on Pattern-Oriented Software Architecture for Concurrent and Networked Software," WaveFront forum at the SPLASH 2013 conference, Indianapolis, IN, October 29th, 2013.
4. "Addressing the Challenges of Tactical Information Management in Net-Centric Systems with the OMG Data Distribution Service (DDS)," the 16th International ACM Workshop on Parallel and Distributed Real-Time Systems (WPDRTS '08), Miami, Florida, April 14, 2008.
5. "The Design and Performance of Configurable Component Middleware for End-to-End Adaptation of Distributed Real-time Embedded Systems," proceedings of the 10th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC), May 7-9, 2007, Santorini Island, Greece.
6. "A Decision-Theoretic Planner for DRE Systems," 7th OMG Real-time/Embedded CORBA workshop, Arlington, VA, July 10–13, 2006.
7. "Model-driven QoS Provisioning for Real-time CORBA and CCM DRE Systems," 6th OMG Real-time/Embedded CORBA workshop, Arlington, VA, July 11–14, 2005.
8. "Research Advances in Middleware for Distributed Systems: State of the Art," Computer Communications stream of the 17th IFIP World Computer Congress, Montreal, Canada, August 25-30, 2002.
9. "Towards Highly Configurable Real-time Object Request Brokers," the IEEE International Symposium on Object-Oriented Real-time Distributed Computing (ISORC), Washington DC, April 29 – May 1, 2002.
10. "Operating System Performance in Support of Real-time Middleware," Proceedings of the 7th IEEE Workshop on Object-oriented Real-time Dependable Systems, San Diego, CA, January, 2002.
11. "Designing an Efficient and Scalable Server-side Asynchrony Model for CORBA," Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001), Snowbird, Utah, June 18, 2001.
12. "DOORS: Towards High-performance Fault-Tolerant CORBA," Proceedings of the 2nd International Symposium on Distributed Objects and Applications (DOA '00), OMG, Antwerp, Belgium, September 2000.
13. "The Design and Performance of a CORBA Audio/Video Streaming Service," Proceedings of the 31st Hawaii International Conference on System Systems (HICSS), Hawaii, January, 1999, mini-track on Multimedia DBMS and the WWW, Hawaii, January 1999.
14. "Alleviating Priority Inversion and Non-determinism in Real-time CORBA ORB Core Architectures," Proceedings of the Fourth IEEE Real-Time Technology and Applications Symposium (RTAS), Denver, Colorado, June 3-5, 1998

15. "Optimizing the Performance of the CORBA Internet Inter-ORB Protocol Over ATM," Proceedings of the 31st Hawaii International Conference on System Systems (HICSS), Hawaii, January, 1998. This was selected as the best paper in the Software Technology Track (188 submitted, 77 accepted).
16. "The Double-Checked Locking Pattern," *Proceedings of the 3rd annual Pattern Languages of Programming conference* in Allerton Park, Illinois, September 4-6, 1996.
17. "Acceptor and Connector: Design Patterns for Initializing Network Services," Proceedings of the EuroPLoP '96 conference, Kloster Irsee, Germany, July 10-14, 1996.
18. "Measuring the Performance of Communication Middleware on High-Speed Networks," Proceedings of SIGCOMM '96, ACM, San Francisco, August 28-30th, 1996.
19. "Design and Performance of an Object-Oriented Framework for High-Speed Electronic Medical Imaging," Proceedings of the $2^{nd}$ Conference on Object-Oriented Technologies and Systems (COOTS), USENIX, Toronto, June 18-22, 1996.
20. "A Family of Design Patterns For Flexibly Configuring Network Services in Distributed Systems," Proceedings of the International Conference on Configurable Distributed Systems, IEEE, Annapolis, Maryland, May 6-8, 1996.
21. "Using Design Patterns to Develop High-Performance Object-Oriented Communication Software Frameworks," Proceedings of the $8^{th}$ Annual Software Technology Conference, Salt Lake City, Utah, April 21-26, 1996.
22. "An Object-Oriented Framework for High-Performance Electronic Medical Imaging," Proceedings of the *Very High Resolution and Quality Imaging* mini-conference at the Symposium on Electronic Imaging in the International Symposia Photonics West 1996, SPIE, San Jose, California USA, January 27 - February 2, 1996.
23. "Half-Sync/Half-Async: A Pattern for Efficient and Well-structured Concurrent I/O," *Proceedings of the $2^{nd}$ Pattern Languages of Programs Conference* Monticello, Illinois, September 6-8, 1995.
24. "Acceptor and Connector: Design Patterns for Actively and Passively Initializing Network Services." Workshop on Pattern Languages of Object-Oriented Programs at ECOOP '95, August 7 – 1, 1995, Aarhus, Denmark.
25. "Object-Oriented Components for High-speed Network Programming," *Proceedings of the Conference on Object-Oriented Technologies (COOTS)*, USENIX, June 26-29, 1995 Monterey, California, USA, pp. 21–38.
26. "Experience Using Design Patterns to Evolve Communication Software Across Diverse OS Platforms," *Proceedings of the $9^{th}$ European Conference on Object-Oriented Programming (ECOOP)*, ACM, Aarhus, Denmark, August, 1995,.
27. "Measuring the Performance of Parallel Message-based Process Architectures," *Proceedings of the INFOCOM Conference on Computer Communications*, IEEE, Boston, MA, April, 1995, pp. 624–633.
28. "High-Performance Event Filtering for Dynamic Multi-point Applications," Proceedings of the $1^{st}$ Workshop on High Performance Protocol Architectures (HIPPARCH), INRIA, Sophia Antipolis, France, December, 1994, p 1–8.
29. "Flexible Configuration of High-Performance Object-Oriented Distributed Communication Systems," $9^{th}$ *OOPSLA Conference, invited paper to the Workshop on Flexibility in Systems Software*, ACM, Portland, Oregon, October, 1994, pp. 1–4.
30. "Performance Experiments on Alternative Methods for Structuring Active Objects in High-Performance Parallel Communication Systems," $9^{th}$ *OOPSLA Conference, poster session*, ACM, Portland, Oregon, October, 1994, pp. 1–12.
31. "Measuring the Impact of Alternative Parallel Process Architectures on Communication Subsystem Performance," *Proceedings of the Proceedings of the $4^{th}$ International Workshop on Protocols for High-Speed Networks*, IFIP, Vancouver, British Columbia, August, 1994, pp. 103–118.
32. "Reactor: An Object Behavioral Pattern for Concurrent Event Demultiplexing and Dispatching," *Proceedings of the $1^{st}$ Annual Conference on the Pattern Languages of Programs*, Monticello, Illinois, August, 1994, pp. 1–10.

33. "Experiences with an Object-Oriented Architecture for Developing Dynamically Extensible Network Management Software," *Proceedings of the Globecom Conference*, IEEE, San Francisco, California, November, 1994, pp. 1–7.
34. "Configuring Function-based Communication Protocols for Distributed Applications," *Proceedings of the* $8^{th}$ *International Working Conference on Upper Layer Protocols, Architectures, and Applications*, IFIP, Barcelona, Spain, June 1-3, 1994, pp. 361–376.
35. "The ADAPTIVE Service Executive: An Object-Oriented Architecture for Configuring Concurrent Distributed Communication Systems," *Proceedings of the* $8^{th}$ *International Working Conference on Upper Layer Protocols, Architectures, and Applications*, IFIP, Barcelona, Spain, June 1-3, 1994, pp. 163–178.
36. "ASX: An Object-Oriented Framework for Developing Distributed Applications," *Proceedings of the* $6^{th}$ *C++ Conference*, USENIX, Cambridge, Massachusetts, April, 1994, pp. 200–220.
37. "The Service Configurator Framework: An Extensible Architecture for Dynamically Configuring Concurrent, Multi-service Network Daemons," *Proceedings of the* $2^{nd}$ *International Workshop on Configurable Distributed Systems*, IEEE, Pittsburgh, PA, March 21-23, 1994, pp. 190–201.
38. "Tools for Generating Application-Tailored Multimedia Protocols on Heterogeneous Multi-Processor Platforms," *Proceedings of the* $2^{nd}$ *Workshop on High-Performance Communications Subsystems*, IEEE, Williamsburg, Virginia, September 1-3, 1993, pp. 1–7.
39. "A Framework for Developing and Experimenting with Parallel Process Architectures to Support High-Performance Transport Systems," *Proceedings of the* $2^{nd}$ *Workshop on High-Performance Communications Subsystems*, IEEE, Williamsburg, Virginia, September 1-3, 1993, pp. 1–8.
40. "ADAPTIVE: a Framework for Experimenting with High-Performance Transport System Process Architectures," *Proceedings of the* $2^{nd}$ *International Conference on Computer Communications and Networks*, ISCA, San Diego, California, June 28-30, 1993, pp. 1–8.
41. "ADAPTIVE: A Flexible and Adaptive Transport System Architecture to Support Lightweight Protocols for Multimedia Applications on High-Speed Networks," *Proceedings of the* $1^{st}$ *Symposium on High Performance Distributed Computing*, IEEE, Syracuse, New York, September 9-11, 1992, pp. 174–186.
42. "GPERF: A Perfect Hash Function Generator," *Proceedings of the* $2^{nd}$ *C++ Conference*, USENIX, San Francisco, California, April 9-11, 1990, pp. 87–102.

**Invited Talks**

1. "Architecting the Future of Software Engineering," invited keynote talk at the 16th International Conference on Software Technologies, July 6-8th, 2021.
2. "Challenges of Certifying Adaptive Dynamic Computing Environments," ARLIS Workshop, January 28th, 2021.
3. "Cyber-Security and You: Practicing Safe Surfing on the Internet," the National Active and Retired Federal Employees (NARFE) chapter, Nashville TN, January 13th, 2021.
4. "Challenges of Certifying Adaptive Dynamic Computing Environments," DARPA/SEI Software Engineering Grand Challenges and Future Visions Workshop, December 1st, 2020.
5. "Survelliance Capitalism and You," invited talk at the Southeast Science Boot Camp, Nashville, TN, May 29th, 2019.
6. "Diversify Sponsorship of Your Research: Getting Funding from the Department of Defense," Office of Research Development and Support Workshop, October 22nd, 2018, Nashville, TN.
7. "Survelliance Capitalism and You," invited talk at the Memorizing the Future: Collecting in the 21st Century Conference, Nashville, TN, October 6th, 2018.
8. "Aligning Incentives to Enable More Effective Organic Software Infrastructure for the DoD," DoD Organic Software Infrastructure Workshop, Arlington VA, August 13th, 2018.
9. "The Blockchain: What It is and Why It Matters to Us," Transforming Dermatology in the Digital Era, Memorial Sloan Kettering Cancer Center, October 25, 2018, NY, NY, USA.
10. "Aligning Incentives to Enable Modular Open Software for DoD Combat Systems," Modular Open Systems Summit, May 1st, 2018, Washington DC.

11. “The Blockchain: What It is and Why It Matters to Us,” Society of Women Engineers, Vanderbilt University, March 14th, 2018.
12. “The Blockchain: What It is and Why It Matters to Us,” Invited keynote at the Workshop on Middleware and Applications for the Internet of Things, (co-located with the 2017 Middleware conference in Las Vegas, USA), December 11th and 12th, 2017.
13. “The Blockchain: What It is and Why It Matters,” Vanderbilt University, Nashville, TN, November 28th, 2017.
14. “The Blockchain: What It is and Why It Matters,” INTERFACE Nashville conference, Nashville, TN, August 24th, 2017.
15. “Applying Blockchain to Healthcare Systems,” panel presentation at the Siemens Blockchain Conference, Nuremburg, Germany, May 10th, 2017.
16. “A Primer on Big Data,” Vanderbilt University Board of Trust meeting, April 21st, 2017, Nashville TN.
17. “The Past, Present, and Future of MOOCs and Their Importance for Software Engineering,” Distinguished Lecture, University of Illinois Chicago, Chicago, IL, November 18th, 2016.
18. “Agility-at-Scale for Safety- and Mission-Critical Industrial-Scale Systems,” INFORMS Annual Conference, Nashville, TN November 13th, 2016.
19. “Product Line Architectures for Open System Architectures,” Varian, Winnipeg, Canada, October 7th, 2016.
20. “Agility-at-Scale for Safety- and Mission-Critical Industrial-Scale Systems,” Siemens Architecture Workshop, Tarrytown, NY, September 27th, 2016.
21. “Product Line Architectures for Oncology Treatment Services,” Varian, Palo Alto, CA, September 16th, 2016.
22. “Innovation and Speed: The Rise of Open Systems,” the United States Technology Leadership Council, Reston, VA, August 24th, 2016.
23. “Elastic Software Infrastructure to Support the Industrial Internet,” the Siemens CPS Workshop, Munich, Germany, August 1st, 2016.
24. “Challenges of Certifying Adaptive Dynamic Computing Environments,” Workshop on Safety And Control for AI, Sponsored by the White House Office of Science and Technology Policy and Carnegie Mellon University, Pittsburgh, PA, June 28th, 2016.
25. “Keeping an Unfair Advantage in a Globalized and Commoditized World,” Raytheon Symposium, Tucson, AZ, May 5th, 2016.
26. “Towards Technical Reference Frameworks to Support Open System Architecture Initiatives,” Office of the Secretary of Defense (OSD) System of Systems Engineering Collaborators Information Exchange, December 15th 2015.
27. “Enterprise System of Systems Engineering (SoSE) Integration and Innovation,” presentation at the US Marine Corp Business Management Association meeting, Quantico, VA, December 10th, 2015.
28. “An Architecture Grand Challenge: DOD’s push for Open Systems Architecture,” panel presentation at the Software Solutions Conference, Crystal City, VA, November 17th, 2015.
29. “Elastic Software Infrastructure to Support the Industrial Internet,” the Siemens CPS Workshop, Munich, Germany, September 29th, 2015.
30. “An Overview of Mobile and mHealth Activities at ISIS and Vandy EECS,” Patient Engagement Emerging Technologies, Vanderbilt University, Nashville, TN, August 10, 2015.
31. “Mobile Cloud Computing with Android,” Google I/O, Mercury Intermedia Systems, Nashville, TN, May 28th, 2015.
32. “An Architecture Grand Challenge: DOD’s push for Open Systems Architecture,” panel presentation at the SATURN 2015 Conference, Baltimore, MD, April 27th, 2015.
33. “‘Elastic Software Infrastructure to Support Computing Clouds for Cyber-Physical Systems,” Distinguish Lecture, Texas A&M, April 27th, 2015.

34. “Elastic Software Infrastructure to Support Computing Clouds for Cyber-Physical Systems”, Boeing Distinguished Researcher And Scholar Seminar (B-DRASS) series, March 20th, Huntingon Beach, CA.
35. “Elastic Software Infrastructure to Support Computing Clouds for Cyber-Physical Systems,” Distinguished Lecture, University of California, Irvine, February 27th, 2015.
36. “Elastic Software Infrastructure to Support Computing Clouds for Cyber-Physical Systems,” Varian, Palo Alto, CA, January 15th, 2015.
37. “Keeping an Unfair Advantage in a Globalized and Commoditized World,” Open Architecture Summit, Washington DC, November 4th, 2014.
38. “Proposal for a Professional Masters degree in Computer Science,” invited talk at Vanderbilt University School of Engineering’s Board of Visitor’s meeting, October 10th, 2014.
39. “The Past, Present, and Future of Open System Architecture Initiatives,” invited keynote at the Future Airborne Capabilities Environment meeting, Nashville, TN, September 24th.
40. “Future Proofing Research and Development Investments in a Globalized, Commoditized World,” Boeing Technical Excellence Conference, May 20th, 2014, St. Louis, MO.
41. “Elastic Software Infrastructure to Support the Computing Clouds for Cyber-Physical Systems (CC4CPS),” Securboration Conference, November 12th, 2013, Melbourne, Florida.
42. “The Importance of Automated Testing in Open Systems Architecture Initiatives,” Open Architecture Summit, November 12th, 2013, Washington DC.
43. Conference on Cloud and Mobile Computing, Siemens Corporate Research, Princeton, NJ, November 5th, 2013.
44. “Overview of the Technology Entrepreneurship Task Force,” Innovation, Imagination, and Introductions: A Conversation with Entrepreneurs, Vanderbilt University, October 24th, 2013.
45. “Producing and Delivering a Coursera MOOC on Pattern-Oriented Software Architecture for Concurrent and Networked Software,” Vanderbilt University’s Faculty Senate committee on Strategic Planning and Academic Freedom, October 23rd, 2013.
46. “Elastic Software Infrastructure to Support the Industrial Internet,” RTI Webinar series, October 23rd, 2013.
47. “The Importance of Applying Agility to DoD Software Initiatives,” IEEE Computer Society Lockheed Martin webinar series, October 10th, 2013.
48. “Technology Entrepreneurship Task force: Charter and Progress Update,” Vanderbilt University School of Engineering Board of Visitors meeting, October 4th, 2013.
49. “Stochastic Hybrid Systems Modeling and Middleware-enabled DDDAS for Next-generation USAF Combat Systems,” AFOSR DDDAS PI meeting, Arlington, VA, October 1st, 2013.
50. “Producing and Delivering a Coursera MOOC on Pattern-Oriented Software Architecture for Concurrent and Networked Software,” WithIT seminar, Vanderbilt University, September 12th, 2013.
51. “Applying Agility to the US Department of Defense Common Operating Platform Environment Initiatives,” Interoperable Open Architecture conference, Washington DC, September 11th, 2013.
52. “Software Infrastructure Support of Computing Clouds for Cyber-Physical Systems,” invited talk at Real-Time Innovations, July 31st, 2013, Sunnyvale, California.
53. “Introduction to the Institute for Software Integrated Systems,” Nashville Entrepreneur Center, July 15th, 2013.
54. ”Surviving the Coursera Digital Learning Experience,” Coursera-in-TN Conference, Vanderbilt University, Nashville, TN, June 24th, 2013.
55. “Quo Vadis ISORC?,” Panel presentation at ISORC 2013 Conference, June 19th, 2013, Paderborn, Germany.
56. “Software Infrastructure Support of Computing Clouds for Cyber-Physical Systems,” invited keynote for ISORC 2013 Conference, June 19th, 2013, Paderborn, Germany.
57. “Towards Programming Models and Paradigms for Computing Clouds that Support Cyber-Physical Systems,” NSF Workshop on Computing Clouds for Cyber-Physical Systems, March 15th, 2013, Ballston, VA.

58. “Built to Last: Planning Your Career as an Engineer,” STEM contest on Securing Cyber Space, Brentwood High School, March 9th, 2013, Nashville, TN.
59. “Experience with Digital Learning and MOOCs at Vanderbilt,” Nashville, TN, Feb 22nd, 2013.
60. “Software Design: Is It Really Better to Look Good Than to Feel Good?,” World IA Day, Nashville, TN, Feb 9th, 2013.
61. “Pattern-Oriented Software Architectures: Patterns and Frameworks for Concurrent and Networked Software,” PhreakNIC 2012, Murfreesboro, TN, November 9th, 2012.
62. “Applying Agility to the US Department of Defence Common Operating Platform Environment Initiatives,” Interoperable Open Architecture 2012, 29 - 31 October, 2012, London, UK.
63. “Open System Architectures: Challenges and Success Drivers,” OA Summit conference, Washington, DC, October 18th, 2012.
64. “Dependable Computing Clouds for Cyber-Physical Systems,” Dependability Issues in Cloud Computing Workshop, October 11th, 2012, Irvine, CA.
65. “Computing Clouds for Cyber-Physical Systems,” Reliable Cloud Infrastructure for CPS Applications Workshop, October 8th, 2012, Irvine, CA.
66. “Common Operating Platform Environments: Challenges and Success Drivers,” Navy Open Systems Architecture workshop, Ballston, VA, September 27th, 2012.
67. ”Meeting the Challenges of Enterprise Distributed Real-time and Embedded Systems,” talk for Honeywell Aerospace, September 21, 2012.
68. “Architecture-Led Iterative and Incremental Development for Common Operating Platform Environments,” NITRD Software Design and Productivity meeting, National Coordination Office, Ballston, VA, July 13th, 2012.
69. ”Cyber-physical multi-core Optimization for Resource and cachE effectS,” Software-Intensive Systems Producibility workshop, Arlington VA, June 5th, 2012.
70. ”Applying Agility to DoD Common Operating Platform Environment Initiatives”, SEI Agile Research Forum, May 22nd, 2012.
71. “Meeting the Challenges of Enterprise Distributed Real-time and Embedded Systems,” keynote talk at the SATURN Conference 2012 May 7-11, 2012, St. Petersburg, FL.
72. “Reflections on 20 Years of Architecture for Distributed Real-time and Embedded Systems,” SATURN Conference 2012 May 7-11, 2012, St. Petersburg, FL.
73. “US Naval Open Systems Architecture Strategy,” SATURN Conference 2012 May 7-11, 2012, St. Petersburg, FL.
74. ”Towards Open Systems Architectures for Distributed Real-time and Embedded Systems,” The Center for Embedded Systems for Critical Applications, Annual Workshop, Virginia Tech, Blackburg, VA April 21st, 2012.
75. “Overview of the SEI Strategic Research Plan,” ASD(R&E) Annual Program Review, December 7th, 2011, Pittsburgh, PA.
76. “Overview of the SEI Strategic Research Plan,” Acquisition Support Program meeting, November 16th, 2011, Pittsburgh, PA.
77. “Conducting Leading-Edge Software R&D in a Globalized, Commoditized World,” NITRD Software Design and Productivity meeting, National Coordination Office, Ballston, VA, November 3rd, 2011.
78. “A Technical Assessment of Open Architecture Systems for Military Use,” Interoperable Open Architecture, 26th-28th October 2011, London, UK.
79. “Conducting Leading-Edge Software R&D in a Globalized, Commoditized World,” Technovation 2011, Carnegie Mellon University, September 29th, 2011.
80. “CTO Report,” SEI Board of Visitors Meeting, Arlington, VA, September 27th, 2011.
81. “Overview of the SEI Strategic Research Plan,” Joint Advisory Committee Meeting, Arlington, VA, September 26th, 2011.

82. "Successful Development Efforts: Standards, People, & Culture: The Enterprise Perspective," Software Assurance (SwA) Forum, September 16th, 2011, Arlington, VA.
83. "Ultra-Large-Scale (ULS) Cyberphysical Systems and Their Impact on Technology and Society," University of Salzburg, June 30th, 2011, Salzburg, Austria.
84. "Ultra-Large-Scale (ULS) Cyberphysical Systems and Their Impact on Technology and Society," ARTEMIS conference, June 29th, 2011, Linz, Austria.
85. "Ultra-Large-Scale Systems and Their Impact on the DoD," Systems and Software Technology Conference Committee, keynote presentation at the 23rd Systems and Software Technology Conference, May 16-19, 2011, Salt Lake City, Utah.
86. "Ultra-Large Scale Systems and their Impact on Technology and Society," keynote presentation at the International Symposium on Object-Oriented Real-time Distributed Computing¡/A¿ (ISORC), Newport Beach, CA, March 29th, 2011.
87. "Software-reliant Systems Research at the Software Engineering Institute," Raytheon, Sudbury, MA, March 10, 2011.
88. "Review of COE Practices," US Army Senior Leadership Education Program, Pittsburgh, PA, January 20th, 2011.
89. "Software Producibility for Defense," US Army Senior Leadership Education Program, Pittsburgh, PA, January 18th, 2011.
90. "SEI Research: The Shape of Things to Come," ASP Meeting, Software Engineering Institute, Pittsburgh, PA, December 9th, 2010.
91. "R&D at ASP," ASP Air Force Training Day, Software Engineering Institute, Pittsburgh, PA, December 9th, 2010.
92. "Software-reliant Systems Research at the Software Engineering Institute," Siements Corporate Research, Princeton, NJ, November 22nd, 2010.
93. "Taming the Complexity of Software-Reliant Systems," Software Engineering Process Group conference, Colombia, South America, November 11th, 2010.
94. "SEI Technical Presentations," Joint Advisory Committee Meeting, Arlington, VA, October 26th, 2010.
95. "SEI Research: The Shape of Things to Come," ASP Meeting, Software Engineering Institute, Pittsburgh, PA, October 20th, 2010.
96. "SEI Research: The Shape of Things to Come," SEPM Meeting, Software Engineering Institute, Pittsburgh, PA, October 19th, 2010.
97. "Strategic Directions for Research at the SEI," RTSS Offsite Meeting, Pittsburgh, PA, October 12th, 2010.
98. "The World is Flat and What You Can Do About It," Family Weekend, October 9th, 2010, Vanderbilt University.
99. "SEI Research: The Shape of Things to Come," SEI Board of Visitor's Meeting, Arlington, VA, September 28th, 2010.
100. "SEI Research: The Shape of Things to Come," PD&T Meeting, Software Engineering Institute, Pittsburgh, PA, September 20th, 2010.
101. "Introduction and Initial Thoughts," RTSS Meeting, Software Engineering Institute, Pittsburgh, PA, August 19th, 2010.
102. "The Impact of Ultra-Large-Scale Systems on DoD Operations," Congressional R&D Caucus, Rayburn Building, Washington DC, January 19th, 2010.
103. "The World is Flat and What You Can Do About It," Explorers meeting, January 12th, 2010, Vanderbilt University.
104. "Expectations for University - Industry Collaborative Research in CPS," Computing Community Consortium Workshop on New Forms of Industry-Academy Partnerships in CPS Research, George Mason University, May 19th, 2009.
105. "How Good is Your SOA?", Panel presentation at the AFRL QED PI meeting, April 28th, 2009, Washington DC.

106. "The World is Flat and What You Can Do About It," ES 140, Computer Science module, October 31st, 2008, Vanderbilt University.
107. “Meeting the Challenges of Ultra-Large-Scale Distributed Real-time and Embedded Systems with QoS-enabled Middleware and Model-Driven Engineering,” Panel on Growing and Sustaining Ultra Large Scale (ULS) Systems, OOPSLA 2008, Nashville TN, October 21-23 2008.
108. "The World is Flat and What You Can Do About It," Family Weekend Faculty Lecture, Vanderbilt University, October 3rd, 2008.
109. “The World is Flat and What You Can Do About It,” Senior Design Seminar, Vanderbilt University, September 17th, 2008.
110. “The World is Flat and What You Can Do About It,” CS WithIT Seminar, Vanderbilt University, September 11th, 2008.
111. "The Managed Motorway: Real-time Vehicle Scheduling - A Research Agenda," Qualcomm, July 28th, 2008, San Diego, CA.
112. "Meeting the Challenges of Mission-Critical Distributed Event-Based Systems with QoS-enabled Middleware and Model-Driven Engineering," 2nd International Conference on Distributed Event-Based Systems (DEBS), Rome Italy, July 2-4, 2008.
113. "Meeting the Challenges of Distributed Real-time and Embedded Systems with QoS-enabled Middleware and Model-Driven Engineering," SPAWAR, April 29th, 2008.
114. "Meeting the Challenges of Distributed Real-time and Embedded Systems with QoS-enabled Middleware and Model-Driven Engineering," Northrop Grumman, Boulder Colorado, April 25th, 2008.
115. “Experimentation Environment for QED,” AFRL Information Management PI Meeting, April 16 2008, Georgetown, Washington, DC.
116. “Adaptive System Infrastructure for Ultra-Large-Scale Systems,” SMART Conference, Carnegie Mellon University, March 6th, 2008.
117. "Experimentation Environment for QED", Air Force Research Lab, Rome, NY, March 4th, 2008.
118. "Ultra-Large-Scale (ULS) Systems and their Impact on Technology and Society," Clemson University, January 31st, 2008.
119. "Meeting the Challenges of Ultra-Large-Scale Distributed Real-time and Embedded Systems with QoS-enabled Middleware and Model-Driven Engineering, invited keynote talk at Middleware 2007, Irvine, CA, November 29th, 2007.
120. "The World is Flat and What You Can Do About It," Senior Design Seminar, Vanderbilt University, November 14th, 2007.
121. “Technology Candidates for QED,” AFRL retreat, Minnowbrook, NY, October 23, 2007.
122. "Overview of ISIS and Proposed IU/CRC R&D Projects," Crystal City, VA, October 19th, 2007.
123. The Future of CORBA for Distributed Real-time and Embedded Systems, International Conference on Accelerator and Large Experimental Physics Control Systems, October 17, 2007, Knoxville, TN.
124. "AF-TRUST: Project Overview," Air Force Scientific Advisory Board review, Rome, NY, October 15th, 2007.
125. "Meeting the Challenges of Distributed Real-time and Embedded Systems with Product-Line Architectures," August 1st, 2007, Trinity College, Dublin, Ireland.
126. "Model Driven Engineering of Product-Line Architectures for Distributed Real-time and Embedded Systems," July 5th, 2007, University of Limerick, Ireland.
127. "Meeting the Challenges of Mission-Critical Systems with Middleware and Model Driven Engineering", OMG Technical Meeting, June 27, 2007, Brussels, Belgium.
128. Meeting the Challenges of Ultra-Large-Scale Distributed Real-time and Embedded Systems with Model-Driven Engineering, June 19, 2007, Trinity College, Dublin.
129. Strategic Technology Positioning, PrismTechnologies "Middleware Fest", June 14, 2007, Newcastle, UK.
130. "Hurdles for Wireless Communication Systems R&D and Some Ways to Overcome Them," OSD Workshop on Wireless Communication Systems, Rosslyn, VA, May 22nd, 2007.

131. “The World is Flat from a Computer Scientists Point of View,” Vanderbilt University Commencement talk, May 10th, 2007.
132. Meeting the Challenges of Ultra-Large-Scale Distributed Real-time and Embedded Systems, invited keynote at the the 10th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing, May 7-9, 2007, Santorini Island, Greece.
133. “Enhanced QoS for the GIG,” AFRL JBI PI meeting, Georgetown, DC, April 24, 2007.
134. “Meeting the Challenges of Ultra-Large-Scale Distributed Real-time and Embedded Systems,” Invited keynote at the 15th International Workshop on Parallel and Distributed Real-Time Systems (WDPRTS), March 26-27, 2007, Long Beach, California.
135. “The CORBA C++ Mapping: Beyond Repair?,” OMG Meeting, San Diego, CA, March 27th, 2007.
136. “Meeting the Challenges of Ultra-Large-Scale Systems via Model-Driven Engineering,” Distinguished Lecturer Series, Florida International University, Miami, Florida, Feb 2, 2007.
137. Model Driven Engineering and QoS-enabled Component Middleware for DRE Systems, Invited talk at the European Space Agency Operations Center, Darmstadt, Germany, Wednesday January 24, 2007.
138. ”Software Wind Tunnel (SWiT) Concept of Operations and System Architecture”, AFRL Software and Systems Test Track workshop, Arlington, VA, January 19, 2007.
139. “Latest Breakthroughs in SDR Software Development Using Model Driven Technologies,” Rockwell Collins, Cedar Rapids, IA, December 14th, 2006.
140. “Educating the DoD Workforce in a Flat World,” 2006 Raytheon Integrated Defense Systems’ SW Engr. Directorate Off-Site Meeting, New Castle, New Hampshire, December 7, 2006.
141. “The Ultra Challenge: Software Systems Beyond Big,” panelist at OOPSLA 2006, October, 2006, Portland, OR.
142. “Software Wind Tunnel (SWiT) Architecture,” AFRL Software and Systems Test Track Workshop, Cherry Hill, NJ, October 2nd, 2006.
143. “The World is Flat and What You Can Do About it,” Vanderbilt University, September 12th, 2006.
144. “The World is Flat and What You Can Do About it,” Vanderbilt University, September 8th, 2006.
145. “Meeting the Challenges of Ultra-Large-Scale Systems via Model-Driven Engineering,” Network-Centric Operations Industry Consortium, Reston, VA, August 2nd 2006.
146. Model Driven Architecture Roundtable, invited panelist at the Software Engineering Institute, Pittsburgh, PA, June 1st, 2006.
147. “Enhanced QoS for the GIG,” AFRL JBI PI meeting, Tysons Corner, VA, April 11, 2006.
148. “Model Driven Engineering for Distributed Real-time and Embedded Systems,” Distinguished Lecturer Series talk at Colorado State University, Ft. Collins, CO, April 10, 2006.
149. ”Win-Win Partnership of Academia and Industry: Why Should We Care? Where Is Our Common Future?” invited panelist at the 12th IEEE Real-Time and Embedded Technology and Applications Symposium April 6, 2006, San Jose, California.
150. “Meeting the Challenges of Ultra-Large-Scale Real-time Systems,” invited keynote at the IEEE Real-Time and Embedded Technology and Applications Symposium April 5, 2006, San Jose, California.
151. “Model-driven Development for Distributed Real-time and Embedded Systems,” ACM Meeting at Middle Tennessee State University, March 7th, 2006.
152. “Real-time, Scalable, and Secure Information Management for the GIG,” Scientific Advisory Board Meeting, Rome, NY, November 16th, 2005.
153. “Real-time, Scalable, and Secure Information Management for the GIG,” Airforce Research Lab, Rome, NY, November 3rd, 2005.
154. “Model-driven Development for Distributed Real-time and Embedded Systems,” Distinguished Speaker Talk at BBN Technologies, Cambridge, MA, October 27, 2005.

155. "Challenges and Research Areas for QoS-enabled Information Management in Tactical Systems of Systems," AFRL Minnowbrook Workshop, Adirondack Mountains, NY, October 21st, 2005.
156. "Model-driven Development for Distributed Real-time and Embedded Systems," Invited keynote at MODELS 2005, ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Half Moon Resort, Montego Bay, Jamaica, October 5-7, 2005.
157. "The World is Flat and What You Can Do About it," CS WithIT Seminar, Vanderbilt University, September 22, 2005.
158. "Why Software Reuse has Failed and How to Make it Work for You," Motorola 2005, Symposium on Software, Systems, and Simulation, Schaumburg, IL, September 16th, 2005.
159. "Pattern-Oriented Software Architecture," 12th Pattern Language of Programming Conference, Allerton Park, Illinois, September 7-10, 2005.
160. "Model-Driven Development of Distributed Real-time and Embedded Systems," 12th Pattern Language of Programming Conference, Allerton Park, Illinois, September 7-10, 2005.
161. "Model-driven Development for Distributed Real-time and Embedded Systems," Siemens Corporate Research, Princeton, NJ, August 26th.
162. "Model-driven QoS Provisioning for Real-time CORBA and CCM DRE Systems," 6th OMG Real-time/Embedded CORBA workshop, Washington DC, July 11–14, 2005.
163. "A Proposed R&D Agenda for the Software Technology Laboratory," Lockheed Martin Advanced Technology Lab, Cherry Hill, NJ, June 28th, 2005.
164. "Model-Driven Development of Product-Line Architectures for DRE Systems," 11th Siemens Software Architecture Improvement Group (SAIG), Buffalo Grove, IL June 22, 2005.
165. "Business Drives for Platforms," panel at the 11th Siemens Software Architecture Improvement Group (SAIG), Buffalo Grove, IL June 22, 2005.
166. "Model Driven Development for Distributed Real-time and Embedded Systems," Lockheed Martin Advanced Technology Lab, Cherry Hill, NJ, June 15th, 2005.
167. "Approaches for Supporting Real-time QoS in JBI," JBI PI Meeting, Washington DC, May 24th, 2005.
168. "Overcoming Hurdles of Software Producibility," OSD, Software Producibility Workshop, Arlington, VA, May 18, 2005.
169. "Overview of Multi-Level Resource Management in ARMS," Fermilab, Chicago, IL, April 12th, 2005.
170. "Model Driven Middleware for Distributed Real-time and Embedded Systems," University of Southern Alabama, April 8, 2005.
171. "Model-Driven Development of Distributed Real-time and Embedded Systems," UAV Battlelab, Indian Springs, NV, February 10th, 2005.
172. "The Future of Software and Systems Engineering," IEEE Meeting, Vanderbilt University, February 8th, 2005.
173. Model Driven Development of Distributed Real-time and Embedded Systems, panel at the OOP conference, Munich, Germany, January 27, 2005.
174. "Product-line Architecture Technologies for Distributed Real-time and Embedded Systems, Lockheed Martin, Moorestown, NJ, November 11, 2004.
175. "Model Driven Development of Distributed Real-time and Embedded Systems," invited panelist in the "Generative Programming: Past, Present, and Future," at the 3rd ACM International Conference on Generative Programming and Component Engineering, Vancouver, CA, October 24th 2004.
176. "Developing Combat Systems with Component Middleware and Models," Lockheed Martin, Moorestown, NJ, October 22, 2004.
177. "Model Driven Development of Distributed Real-time and Embedded Systems," Lockheed Martin Advanced Technology Lab, Cherry Hill, NJ, October 21, 2004.
178. "Model Driven Development of Distributed Real-time and Embedded Systems," Lockheed Martin Missile and Fire Control, Dallas, TX, October 13, 2004.

179. “Design of ARMS MLRM Components: CCM Based Design for Dynamic Resource Management,” DARPA ARMS Technical Interchange Meeting, Plymouth, RI, October 7, 2004.
180. “Model Driven Middleware for Component-based Distributed Systems,” keynote for the The 8th International IEEE Enterprise Distributed Object Computing Conference, Monterey, California, September 22, 2004.
181. “Systems Science Challenge Area,” TRUST NSF Science and Technology Review, UC Berkeley, September 12, 2004.
182. “Model Driven Development for Distributed Real-time and Embedded Systems,” Lockheed Martin, Eagan, MN, August 31st, 2004.
183. “Model Driven Computing for Distributed Real-time and Embedded Systems,” Telcordia, Piscataway, NJ, August 10th, 2004.
184. “Model Driven Computing for Distributed Real-time and Embedded Systems,” Raytheon, Portsmouth, RI, August 9th, 2004.
185. “Distributed Object Computing with CORBA,” Raytheon, Portsmouth, RI, August 9th, 2004.
186. “Model Driven Development of Distributed Real-time and Embedded Systems,” Raytheon, Ft. Wayne, IN, July 27th, 2004.
187. “Model Driven Middleware for Distributed Real-time and Embedded Systems,” panelist at the 5th OMG Real-time and Embedded Middleware Workshop, Reston, VA 2004.
188. “The Role of Open Standards,Open-Source Development, and Different Development Models and Processes on Industrializing Software,” ARO Workshop on Software Reliability for FCS, Vanderbilt University, Nashville, Tennessee, May 18-19, 2004.
189. “Model Driven Middleware for Distributed Real-time and Embedded Systems,” Keynote talk for the SIGS Software Engineering Today conference in Zurich, Switzerland, May 4-5, 2004.
190. “Model-Driven Development of Distributed Real-time and Embedded Systems,” 10th Siemens Software Architecture Improvement Group (SAIG), Vienna, Austria, April 20-24, 2004.
191. “Adaptive and Reflective Middleware for Distributed, Real-time, and Embedded Systems,” Purdue University, West Lafayette, Indiana, April 6, 2004.
192. “Model Driven Middleware for Distributed Real-time and Embedded Systems,” *Technologies That Will Change the World* session at the Southeastern Software Engineering Conference, Huntsville, Alabama, March 30th, 2004.
193. “Advances in COTS Middleware for Distributed Real-time and Embedded Systems,” Keynote for the International Conference on COTS-Based Software Systems (ICCBSS) 2004 in Redondo Beach, February 2-4, 2004.
194. Composable Middleware Components for High Confidence Network Embedded Systems, University of California, Berkeley, December 4th, 2003.
195. “Model Driven Middleware,” TechConnect 2003, St. Louis, MO, October 1st, 2003.
196. “Advances in Model Driven Middleware for Distributed Real-time and Embedded Systems,” the Model Integrated Computing PSIG meeting at the OMG Technical Meeting, September 10, 2003, Boston, MA.
197. Invited panelist for the “Research on DRE Systems” panel at the OMG Real-time Middleware Workshop, July 16, 2003, Arlington, VA.
198. “Advances in Model Driven Middleware for Distributed Real-time and Embedded Systems,” the OMG Real-time Middleware Workshop, July 15, 2003, Arlington, VA.
199. Organizer and presenter for a panel on “Advances in Large-scale Distributed Real-time and Embedded Systems” at the 9th IEEE Real-time/Embedded Technology and Applications Symposium (RTAS), May 27-30, 2003, Washington, DC.
200. “Managing Project Risk for Combat Systems,” The Southeastern Software Engineering Conference, Huntsville, Alabama, April 1st, 2003.
201. “Distributed Real-time and Embedded Systems at DARPA,” OMG Workshop on Super Distributed Objects, Washington DC, Monday, November 18, 2002.

202. “Adaptive and Reflective Middleware for Distributed Real-time Systems,” Workshop on High Performance, Fault Adaptive, Large Scale Real-time Systems, Vanderbilt University, November 14, 2002.
203. Invited panelist on ”Objects and Real-time Systems” OOPSLA ’02, Seattle, WA, November 8, 2002.
204. “An Overview of ACE+TAO,” Boeing, Seattle, November 8th, 2002.
205. “Pattern-Oriented Software Architecture,” Amazon, Seattle, WA, November 6th, 2002.
206. “Using Real-time CORBA Effectively: Patterns and Principles,” CORBA Controls Workshop, Grenoble, France, October, 9th, 2002.
207. “Adaptive and Reflective Middleware for Distributed Real-time and Embedded Systems,” EMSOFT 2002: Second Workshop on Embedded Software, Grenoble, France, October, 7–9th, 2002.
208. “Designing the Future of Embedded Systems at DARPA IXO,” Keynote talk at the 6th Annual Workshop on High-Performance Embedded Computing (HPEC), September 25, Boston, MA.
209. “Open Distributed Computing Platforms,” NSF/OSTP Workshop on Information Technology Research for Critical Infrastructure Protection, Lansdowne, VA, September 20th, 2002.
210. “Real-time Object-Oriented Middleware,” Distributed Common Ground/Surface System Technical Review Group meeting, Mclean VA, September 19th, 2002.
211. “Research Advances in Middleware for Distributed, Real-time, and Embedded Systems,” Computer Communications stream of the 17th IFIP World Computer Congress, Montreal, Canada, August 25-30, 2002.
212. “DARPA Thrusts in Embedded Computing,” Mercury Computer Systems, Tyngsboro, MA, July 25th, 2002.
213. “Adaptive and Reflective Middleware for Distributed, Real-time, and Embedded Combat Systems,” Boeing Space and Missile Systems, Anaheim, CA, July 9, 2002.
214. “Annual Report on Software Design and Productivity Coordinating Group,” Interagency Working Group, ITR&D Spring Planning Meeting, NSF, Ballston, VA, May 10, 2002.
215. “Real-time CORBA Standardization: Past, Present, and Future,” panelist in the ”Standards Movements in Object-oriented Real-time Computing” panel at the ISORC 2002 Conference, Washington, DC, April 30, 2002.
216. “Towards Adaptive and Reflective Middleware for Distributed Real-time Embedded Systems,” Moderator of the *Distributed, Real-time, and Embedded Middleware for Network-Centric Combat Systems* panel at the Software Technology Conference (STC) in Salt Lake City, Utah, April 29, 2002.
217. “Applying Architectural Patterns to Address Key Challenges of Distributed Software,” Siemens Architecture Interworking Group, Chicago, IL, April 24, 2002.
218. “Towards Adaptive and Reflective Middleware for Distributed Real-time and Embedded Systems,” Space and Missile Defense Command, Huntsville, AL, April 22, 2002.
219. “How to Maintain Superiority in the Face of the Commoditization of IT,” tutorial at the UCI CEO Roundtable, Maui, Hawaii, April 12, 2002.
220. “Transformation or Transmogrification? Surviving the Commoditization of IT,” panelist at the UCI CEO Roundtable, Maui, Hawaii, April 11, 2002.
221. “Patterns and Principles of Mission-critical Middleware,” Henry Samueli School of Engineering Research Review, University of California, Irvine, March 14th, 2002.
222. “DARPA: an Agency Overview,” CRA Academic Careers Workshop, Arlington, Virginia, February 10 - 12, 2002.
223. “Towards Adaptive and Reflective Middleware for Distributed, Real-time, and Embedded Systems,” Electrical Engineering and Computer Science Department, Vanderbilt University, January 28th, 2002.
224. “Protecting Critical Cyber Infrastructure from Asymmetric Threats,” panelist at the 7th IEEE Workshop on Object-oriented Real-time Dependable Systems, San Diego, CA, January 10, 2002.

225. “The Researcher’s Dilemma: When Technology Success Causes Great Communities to Fail (at Mission-oriented R&D Agencies),” Software Design and Productivity Coordinating Group Workshop on New Visions for Software Design and Productivity: Research and Applications, Nashville, TN, December 13-14, 2001.
226. “Towards Adaptive and Reflective Middleware for Mission-Critical Systems,” Computer Science Department, College of William and Mary, September 7th, 2001.
227. “Adaptive and Reflective Middleware Systems,” Lockheed Martin, Moorestown, NJ, August 21st, 2001.
228. “Adaptive and Reflective Middleware Systems,” United Technology Research Center, Hartford, Connecticut, June 28th, 2001.
229. “Adaptive and Reflective Middleware Systems,” Raytheon Annual Processing Systems Technology Network (PSTN) Symposium, Lexington, MA, June 20th, 2001.
230. Invited presenter for the Vendors’ Panel at the OMG 2nd Workshop on Real-time and Embedded Distributed Object Computing, June 4-7, 2001.
231. ”Towards Pattern Languages and QoS-enabled Middleware for Distributed Real-time and Embedded Systems,” DARPA ITO workshop on Embedded Software, Lake Tahoe, NV, October 8-10, 2001.
232. ”TAO, CORBA, and the HLA/RTI”, Keynote talk at the Fifth IEEE International Workshop on Distributed Simulation and Real Time Applications Cincinnati, Ohio, USA August 13-15, 2001.
233. “Patterns and Principles of Middleware for Distributed Real-time and Embedded Systems,” Raytheon, Sudbury, March 29th, 2001.
234. “Adaptive and Reflective Middleware Systems,” Distinguished Lecture at Florida Atlantic University, Boca Raton, FL, March 1st, 2001.
235. “Adaptive and Reflective Middleware for Mission-Critical Distributed and Embedded Systems,” University of Alabama, Birmingham, AL, January 31st, 2001.
236. “Adaptive and Reflective Middleware for Mission-Critical Distributed and Embedded Systems,” Telcordia, Morristown, NJ, November 20th, 2000.
237. “Adaptive and Reflective Middleware for Mission-Critical Distributed and Embedded Systems,” George Mason University, Fairfax, VA, November 20th, 2000.
238. “Adaptive and Reflective Middleware for Mission-Critical Distributed and Embedded Systems,” Lucent CORBA Forum, Naperville, IL, November 17th, 2000.
239. “Putting an ORB on a Diet,” Session on *Performance and QoS of Embedded CORBA ORBs* at the OMG’s Workshop on Embedded Object-Based Systems, January 17-19, 2001.
240. “Adaptive and Reflective Middleware Systems,” Panelist in a session on “Highly Distributed Systems,” at the IEEE Symposium on Applications and the Internet, San Diego, CA, January 10, 2001.
241. “Adaptive and Reflective Middleware Systems,” Panelist at the NSF Networking PI meeting, Irvine California, November 1st, 2000.
242. “Surviving the Tornado: The Best Kept Secrets of R&D Success in the Internet Age,” Keck Observatory, Hawaii, October 9th, 2000.
243. “Adaptive and Reflective Middleware Systems,” BBN Technologies, Boston, MA, September 27th, 2000.
244. “Distributed Application Integration: Myth or Reality?” Keynote talk at 2nd International Symposium on Distributed Objects and Applications (DOA ’00), OMG, Antwerp, Belgium, September 21st, 2000.
245. “Surviving the Tornado: The Best Kept Secrets of R&D Success in the Internet Age,” Keynote talk at 2nd International Symposium on Distributed Objects and Applications (DOA ’00), OMG, Antwerp, Belgium, September 21st, 2000.
246. “High Confidence Adaptive and Reflective Middleware: Fact or Fiction?” Keynote talk for the IFIP Fourth International Conference on Formal Methods for Open Object-Based Distributed Systems, (FMOODS 2000), Stanford University, Stanford, CA, September 7th, 2000.

247. "Adaptive and Reflective Middleware Systems," Lockheed Martin, Ft. Worth, TX, September 6th, 2000.
248. Pattern-oriented Software Architecture: Concurrent and Networked Objects, Raytheon, San Diego, August 25, 2000.
249. "Adaptive and Reflective Middleware Systems," Rockwell/Collins, Cedar Rapids, Iowa, August 22, 2000.
250. "Adaptive and Reflective Middleware Systems," Lockheed Martin, Eagan, MN, August 21, 2000.
251. "Adaptive and Reflective Middleware Systems," Honeywell Technology Center, Minneapolis, MN, August 18, 2000.
252. "Adaptive and Reflective Middleware Systems," Raytheon, Falls Church, VA, July 12, 2000.
253. "Applying Patterns to Develop High-performance and Real-time Object Request Brokers," Lockheed Martin, Eagan, Minnesota, May 19, 2000.
254. "Patterns and Principles of Real-time Object Request Brokers," Cisco, San Jose, April 12, 2000.
255. "Patterns and Principles of Real-time Object Request Brokers," BellSouth, Atlanta, Georgia, March 3, 2000.
256. "Patterns and Principles of Real-time Object Request Brokers," Distinguished Lecturer Series, Michigan State University, East Lansing, Michigan, October 21, 1999.
257. "Towards Minimum ORBs for Wireless Devices and Networks," OPENSIG '99 Workshop, Carnegie Mellon University, Pittsburgh, October, 14-15, 1999.
258. "Applying CORBA Fault Tolerant Mechanisms to Network Management," Lucent CORBA Forum, Naperville, IL, September 28th, 1999.
259. "CORBA for Real-time and Embedded Telecom Systems," Lucent CORBA Forum, Naperville, IL, September 28th, 1999.
260. "Patterns and Principles of Real-time Object Request Brokers," BEA, Munich, Germany, September 16th, 1999.
261. "Real-time CORBA – Fact or Fiction," Siemens CORBA Day, Munich, Germany, September 15th, 1999.
262. "Patterns and Principles of Real-time Object Request Brokers," Siemens MED, Erlangen, Germany, September 13th, 1999.
263. "Patterns and Principles of Real-time Object Request Brokers," RT DII COE TWG, Boeing, Seattle, WA August 25th, 1999.
264. "Patterns for Real-time Middleware," Microsoft, Redmond, WA, August 24th, 1999.
265. "Patterns and Principles of Real-time Object Request Brokers," Lockheed Martin, Eagan, Minnesota, June 22nd, 1999.
266. "Using the ACE Framework and Patterns to Develop OO Communication Software," Dreamworks SGK, Glendale, CA, May 5th, 1999.
267. "Why Telecom Reuse has Failed and how to Make it Work for You," Keynote talk at Nortel Design Forum, Ottawa, CA, April 22nd, 1999.
268. "QoS-enabled Middleware for Monitoring and Controlling High-Speed Networks and Endsystems," Lucent Bell Labs, Murray Hill, NJ, April 15th, 1999.
269. "Optimization Patterns for High-performance, Real-time Object Request Broker Middleware," University of California, Irvine, April, 2nd, 1999.
270. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent, Columbus, OH, March 18-19 and 25-26, 1999.
271. "Using Design Patterns, Frameworks, and Object-Oriented Communication Systems," Lucent, Holmdel, NJ, March 1-4, 1999.
272. Chaired a panel on "Research Directions for Middleware," NSF PI meeting, Washington, DC, January 24th, 1999.
273. "Principles and Patterns of High-performance Real-time CORBA," University of Southern California, Los Angeles, CA, December 10th, 1998.

274. "Real-time CORBA for Telecom – Fact or Fiction?," Bellcore, Morristown, NJ, December 1st, 1998.
275. "Design Patterns for Real-time Object Request Brokers," Silicon Valley Patterns Group, San Francisco, November 15, 1998.
276. "Why Reuse has Failed and how to Make it Work for You," Keynote talk at Lucent Software Symposium, October 27th, Murray Hill, NJ, 1998.
277. "Real-time CORBA – Fact or Fiction," Lucent CORBA Forum, Holmdel, NJ, September 29, 1998.
278. "Applying Software Design Patterns and Framework to Telecommunication Applications," Nortel Advanced Software Computing and Technology, Monday, April 6, 1998, Ottawa, Canada.
279. "Patterns and Performance of Real-time Object Request Brokers," University of California, Santa Barbara, February 20, 1998.
280. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of Frankfurt, Germany, February 12th, 1998.
281. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of Illinois, Urbana-Champaign November 12th, 1997.
282. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of Missouri, Kansas City, October 31st, 1997.
283. "Principles and Patterns of High-performance, Real-time Object Request Brokers," IBM T.J. Watson Research, September 15, 1997.
284. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of California, Santa Barbara, August 21st, 1997.
285. "Principles and Patterns of High-performance, Real-time Object Request Brokers," Lucent Technologies, Naperville, IL August 19th, 1997.
286. "Mastering Software Complexity with Reusable Object-Oriented Frameworks, Components, and Design Patterns," 3rd NSA Software Reuse Symposium, August 20th, 1997.
287. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of Utah, Salt Lake City, Utah, August 11th, 1997.
288. "Using the ACE Framework and Design Patterns to Develop Object-Oriented Communication Software," CERN, Switzerland, July 18th, 1997.
289. "Principles and Patterns of High-performance, Real-time Object Request Brokers," CHOOSE symposium, Zurich, Switzerland, July 17th, 1997.
290. Invited keynote speaker for $2^{nd}$ Component's User Conference, Munich Germany, July 1997.
291. "Principles and Patterns of High-performance, Real-time Object Request Brokers," Lucent Bell Laboratories, Murray Hill, New Jersey, July 9th, 1997.
292. "Using the ACE Framework and Design Patterns to Develop Object-Oriented Communication Software," Lockheed Martin Tactical Systems, Minneapolis, Minnesota, June 26th, 1997.
293. QoS for Distributed Object Computing Middleware – Fact or Fiction?, panel at the Fifth International Workshop on Quality of Service (IWQoS '97), May 22nd, 1997, Columbia University, NYC, USA.
294. "Design Patterns and Frameworks for Developing Object-oriented WWW Clients and Servers," Carleton University, April 11th, 1997.
295. "Principles and Patterns of High-performance, Real-time Object Request Brokers," University of Maryland, College Park, Maryland, April 2nd, 1997.
296. "A High-Performance End system Architecture for Real Time COBRA," SPARTAN Symposium sponsored by US Sprint, Lawrence Kansas, March 18th, 1997.
297. "Experience with CORBA for Communication Systems," Motorola, Chicago, January 24th, 1997.
298. "High-performance CORBA," Bay Area Object Interest Group, Stanford Linear Accelerator Center, California, December 5th, 1996.
299. "Gigabit CORBA – An Architecture for High-performance Distributed Object Computing," Numerical Aerodynamic Simulation group, NASA, Moffett Field, California, December 3rd, 1996.

300. "Towards High-performance, Real-time CORBA," Distinguished Lecturer at Kansas State University, Manhattan, Kansas, November 7th, 1996.
301. "Gigabit CORBA – An Architecture for High-performance Distributed Object Computing," University of California, Los Angeles, October 3rd, 1996.
302. "Design Patterns and Frameworks for Object-Oriented Communication Software," NSA Software Reuse Symposium, August 28th, 1996.
303. "CORBA – the Good, the Bad, and the Ugly," Lucent Bell-Labs, Naperville, IL, August 22nd, 1996.
304. "Components: the Good, the Bad, and the Ugly," keynote talk for the 1st Components Users Conference, SIEMENS, Munich, Germany, July 15th, 1996.
305. "Design Patterns for Object-Oriented Communication Software," IONA Technologies, Ltd, Dublin, Ireland, July 12th, 1996.
306. "OO Design Patterns and Frameworks for Communication Software," Siemens Corporate Research, Princeton, New Jersey, June 27, 1996.
307. "OO Design Patterns for Concurrent, Parallel, and Distributed Systems," IBM Centre for Advanced Studies, North York, Ontario, Canada, June 17, 1996.
308. "Distributed Object Computing with CORBA", Bell Laboratories, Murray Hill, New Jersey, June 11-12th, 1996.
309. "Design Patterns for Object-Oriented Communication Software," Carleton University, Ottawa, Canada, May 21st, 1996.
310. "Integrating LAN-WAN-Celestial Networks with Design Patterns," Featured technical session at the Object World East conference, Boston, MA, May 9th, 1996.
311. "Using Design Patterns to Develop Object-Oriented Communication Software Frameworks and Applications," McMaster's University, Hamilton, Canada, May 2nd, 1996.
312. "Towards Gigabit CORBA – A High-Performance Architecture for Distributed Object Computing," University of Nevada, Reno, April 25th, 1996.
313. "Domain Analysis: From Tar Pit Extraction to Object Mania?" Panelist at the 4th International Conference on Software Reuse, Orlando, Florida, April $25^{th}$, 1996. (other panelists include Spencer Peterson, SEI CMU, Mark Simos, Organon Motives Inc., Will Tracz, Loral, and Nathan Zalman, BNR Inc).
314. "Concurrent Object-Oriented Network Programming with C++," Kodak Imaging Technology Center, April $19^{th}$, 1996.
315. "Using OO Design Patterns and Frameworks to Develop Object-Oriented Communication Systems," INRS/NorTel Workshop on Telecommunication Software, Montreal, CA, March $14^{th}$, 1996.
316. "Concurrent Object-Oriented Network Programming with ACE and C++," for Siemens Medical Engineering, Erlangen Germany, February $15^{th}$, 1996.
317. "OO Componentware" Panelist at the *OOP '96 Conference*, SIGS, Munich, Germany, February $13^{st}$, 1996. (other panelists included Michael Stal (Siemens AG) and Frank Buschmann (Siemens AG).
318. "Using Design Patterns to Develop High-performance Object-Oriented Communication Software Frameworks," for the Department of Information Systems, Institute of Computer Science, Johannes Kepler University of Linz, Austria, February $12^{th}$, 1996.
319. "The Performance of Object-Oriented Components for High-speed Network Programming," for the Digital Libraries research group at Stanford University, Palo Alto California, February $2^{nd}$, 1996.
320. "Distributed Object Computing with CORBA, ACE, and C++," for South Western Bell Telephone advanced distributed systems group, St. Louis, MO., January $26^{th}$, 1996.
321. "OO Design Patterns for Large-Scale Object-Oriented Communication Software Systems," AG Communication Systems, Phoenix, Arizona, December $11 - 13^{th}$, 1995.
322. "Experience Using OO Design Patterns to Develop Large-Scale Object-Oriented Communication Software Systems," Bell Northern Research, 7th Annual Design Forum, Ottawa, Canada, December $6^{th}$, 1995.

323. "Using OO Design Patterns to Develop Large-Scale Distributed Systems," Object Technology International, Ottawa, Canada, November $22^{nd}$, 1995.
324. "Design Patterns for Concurrent, Parallel, and Distributed Systems," North Dallas Society for Object Technology, September $13^{th}$, 1995.
325. "Using Design Patterns for Iridium Communication Services," at Motorola Iridium, Chandler, AZ, June $30^{th}$, 1995.
326. "Object Technology and the World-Wide Information Infrastructure," Panelist at ECOOP '95, Aarhus, Denmark, August $9^{th}$, 1995.
327. "Measuring the Performance of CORBA over ATM Networks," HP Labs, Palo Alto, CA, June $28^{th}$, 1995.
328. "Measuring the Performance of Object-Oriented Components for High-speed Network Programming," The C++ and C SIG user group, New York, New York, June $5^{th}$, 1995.
329. "An Overview of Design Patterns for Object-Oriented Network Programming," St. Louis Chapter of the ACM, St. Louis, MO, March $13^{th}$ 1995.
330. "Design Patterns for Concurrent Object-Oriented Network Programming," Distributed Systems group at Siemens Corporate Research Center, Munich, Germany, March $3^{rd}$, 1995.
331. "Patterns: 'Eureka,' 'Deja-Vu,' or 'Just Say No'?" Panelist at the *OOP '95 Conference*, SIGS, Munich, Germany January $31^{st}$, 1995. (other panelists included Richard Helm, (DMR), Frank Buschmann (Siemens AG), and Dave Thomas (OTI).
332. "Developing Distributed Applications with the ADAPTIVE Communication Environment," *The $12^{th}$ Annual Sun Users Group Conference*, SUG, San Francisco, California, June $17^{th}$, 1994.
333. "Flexible Configuration of High-performance Distributed Communication Systems," presented at the ETH-Zentrum in the Swiss Federal Institute of Technology, Zurich, Switzerland, May $31^{st}$, 1994.
334. "Object Oriented Techniques for Developing Distributed Applications," *Computer Science Department Colloquia*, California State University Northridge, December $7^{th}$, 1993.
335. "Hosting the ADAPTIVE System in the *x*-Kernel and System V STREAMS," *The x-Kernel Workshop*, IEEE, Tucson, Arizona, November $10^{th}$, 1992.
336. "An Environment for Controlled Experimentation on the Performance Effects of Alternative Transport System Designs and Implementations," IBM T. J. Watson Research Center, Hawthorne, New York, September $10^{th}$, 1992.

**Colloquia, Seminars, and Tutorials**

1. "Scalable Reactive Programming with Java," O'Reilly Live Training, January 22nd, 2021.
2. "Programming with Java Lambdas and Streams," O'Reilly Live Training, January 13th, 2021.
3. "Design Patterns in Java," O'Reilly Live Training, November 12th and 13th, 2020.
4. "Design Patterns in Java," O'Reilly Live Training, September 17th and 18th, 2020.
5. "Programming with Java Lambdas and Streams," O'Reilly Live Training, September 14th, 2020.
6. "Core Java Synchronizers," O'Reilly Live Training, August 20th, 2020.
7. "Scalable Reactive Programming with Java," O'Reilly Live Training, August 19th, 2020.
8. "Programming with Java Lambdas and Streams," O'Reilly Live Training, June 1st, 2020.
9. "Design Patterns in Java," O'Reilly Live Training, May 27th and 28th, 2020.
10. "Core Java Synchronizers," O'Reilly Live Training, May 18th, 2020.
11. "Programming with Java Lambdas and Streams," O'Reilly Live Training, March 30th, 2020.
12. "Design Patterns in Java," O'Reilly Live Training, March 23rd and 24th, 2020.
13. "Scalable Concurrency with the Java Executor Framework," O'Reilly Live Training, February 24th, 2020.
14. "Core Java Synchronizers," O'Reilly Live Training, February 10th, 2020.

15. “Design Patterns in Java,” O’Reilly Live Training, January 29th and 30th, 2020.
16. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, January 22nd, 2020.
17. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, January 22nd, 2020.
18. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, November 27th, 2019.
19. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, November 18th, 2019.
20. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, November 6th, 2019.
21. “Design Patterns in Java,” O’Reilly Live Training, November 4th and 5th, 2019.
22. “Design Patterns in Java,” O’Reilly Live Training, September 17th and 18th, 2019.
23. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, September 3rd, 2019.
24. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, August 29th, 2019.
25. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, August 15th, 2019.
26. “Design Patterns in Java,” O’Reilly Live Training, July 29th and 30th, 2019.
27. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, August 15th, 2019.
28. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, July 2nd, 2019.
29. “Design Patterns in Java,” O’Reilly Live Training, June 13th and 14th, 2019.
30. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, May 16th, 2019.
31. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, May 13th, 2019.
32. “Design Patterns in Java,” O’Reilly Live Training, April 17th and 18th, 2019.
33. “Scalable Programming with Java 8 Parallel Streams,” O’Reilly Live Training, March 27th, 2019.
34. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, March 12th, 2019.
35. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, March 5th, 2019.
36. “Design Patterns in Java,” O’Reilly Live Training, February 26th and 27th, 2019.
37. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, February 19th, 2019.
38. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, February 5th, 2019.
39. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, January 22nd, 2019.
40. “Design Patterns in Java,” O’Reilly Live Training, January 7th and 8th, 2019.
41. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, December 11th, 2018.
42. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, December 6th, 2018.
43. “Design Patterns in Java,” O’Reilly Live Training, November 13th and 14th, 2018.
44. “Scalable Concurrency with the Java Executor Framework,” O’Reilly Live Training, October 29th, 2018.
45. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, October 16th, 2018.
46. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, October 4th, 2018.
47. “Design Patterns in Java,” O’Reilly Live Training, September 18th and 19th, 2018.
48. “Programming with Java 8 Lambdas and Streams,” O’Reilly Live Training, September 4th, 2018.
49. “Reactive Programming with Java 8 CompletableFutures,” O’Reilly Live Training, August 30th, 2018.

50. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, August 20th, 2018.
51. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, July 25th, 2018.
52. "Design Patterns in Java," O'Reilly Live Training, July 2nd and 3rd, 2018.
53. "Reactive Programming with Java 8 CompletableFutures," O'Reilly Live Training, June 26th, 2018.
54. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, June 25th, 2018.
55. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, June 8th, 2018.
56. "Design Patterns in Java," O'Reilly Live Training, May 24th and 25th, 2018.
57. "Reactive Programming with Java 8 CompletableFutures," O'Reilly Live Training, April 26th, 2018.
58. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, April 17th, 2018.
59. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, April 13th, 2018.
60. "Design Patterns in Java," O'Reilly Live Training, April 3rd, 2018.
61. "Reactive Programming with Java 8 CompletableFutures," O'Reilly Live Training, March 13th, 2018.
62. "Scalable Programming with Java 8 Parallel Streams: Part 2," O'Reilly Live Training, March 7th, 2018.
63. "Scalable Programming with Java 8 Parallel Streams: Part 1," O'Reilly Live Training, March 6th, 2018.
64. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, March 1st, 2018.
65. "Reactive Programming with Java 8 CompletableFutures," O'Reilly Live Training, February 13th, 2018.
66. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, February 6th, 2018.
67. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, February 1st, 2018.
68. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, January 12th, 2018.
69. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, January 10th, 2018.
70. "Reactive Programming with Java 8 CompletableFutures," O'Reilly Live Training, January 9th, 2018.
71. "Reactive Programming with Java 8 Completable Futures," O'Reilly Live Training, October 23rd, 2017.
72. "Programming with Java 8 Lambdas and Streams," O'Reilly Live Training, October 19th, 2017.
73. "Scalable Programming with Java 8 Parallel Streams," O'Reilly Live Training, October 17th, 2017.
74. "Java 8 Concurrency," O'Reilly Live Training, September 7-8th, 2017.
75. "Java 8 Concurrency," O'Reilly Live Training, August 30-31st, 2017.
76. "Java 8 Concurrency," O'Reilly Live Training, June 28-29th, 2017.
77. "The C++ Standard Template Library," Qualcomm, San Diego, February 16-19, 2016.
78. "The C++ Standard Template Library," Qualcomm, San Diego, October 13-16, 2015.
79. "The C++ Standard Template Library," Qualcomm, San Diego, October 13-16, 2015.
80. "Pattern-Oriented Java Concurrency," InformIT Webinar, May 14th, 2015.
81. "Pattern-Oriented Concurrent Programming with Java," OOP Conference, Munich, Germany, January 30th, 2015.
82. "Concurrent Programming in Android," OOP Conference, Munich, Germany, January 29th, 2015.
83. "The C++ Standard Template Library," Qualcomm, San Diego, October 14-17, 2014.
84. "The C++ Standard Template Library," Qualcomm, San Diego, August 5-8, 2014.
85. "Pattern-Oriented Software Architecture for Concurrent and Networked Software," July 28-31, 2014.
86. "The C++ Standard Template Library," Qualcomm, San Diego, August 5-8, 2014.

87. “The C++ Standard Template Library,” Qualcomm, India, March, 2014.
88. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, January 23-34, 2014.
89. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, October 16-17th, 2013.
90. “Patterns and Frameworks for Concurrent and Networked Software,” 2013 International Summer School on Trends in Computing Tarragona, Spain, July 25-26, 2013.
91. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, January 23-24th, 2013.
92. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, October 4-5th, 2012.
93. “Embedded Systems Patterns for C Developers,” Qualcomm, San Diego, CA, August 28th, September 11th, September 25th, October 9th, October 23rd, and November 6th, 2012.
94. “Embedded Systems Patterns for C Developers,” Qualcomm, San Diego, CA, August 14-15th, 2012.
95. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, May 15-18th, 2012.
96. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, January 25-26th, 2012.
97. “Object-Oriented Software Patterns and Frameworks,” Qualcomm, San Diego, CA, October 11-12th, 2011.
98. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, May 11-12th, 2011.
99. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, January 25-26, 2011.
100. “Pattern-Oriented Software Architecture: A Pattern Language for Concurrent and Networked Software,” SPLASH 2010, October 17-21, 2010, Reno, Nevada.
101. “Pattern-Oriented Software Architectures - Patterns and Frameworks for Concurrent and Networked Software,” ProObject, Hanover, MD, August 11th, 2010.
102. “Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Embedded Systems,” Qualcomm, Bangalore, India, June 21-22, 2010.
103. “Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Embedded Systems,” Qualcomm, Hyderabad, India, June 24-25, 2010.
104. ”Pattern-Oriented Software Architecture: A Pattern Language for High Quality and Affordable Distributed Computing Systems,” IEEE Webinar Series, June 10th, 2010.
105. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, May 12-13, 2010.
106. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, December 16-17, 2009.
107. ”Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing,” OOPSLA 2009, Orlando, FL, October, 2009.
108. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, September 15-16, 2009.
109. “Networked Embedded Systems Patterns for C Developers,” Qualcomm, San Diego, CA, June 11-12, 2009.
110. ”Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing,” Software Architecture Technology Users’ Network (SATURN) workshop May 5, 2009 in Pittsburgh, PA.
111. “The C++ Standard Template Library,” Qualcomm, San Diego, CA, January 29-30, 2009.
112. ”Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing,” IEEE Webinar Series, January 8th, 2009.
113. ”Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing,” OOPSLA 2008, Nashville, TN, October 20, 2008.
114. ”The Data Distribution Service for Real-time Systems,” OOPSLA 2008, Nashville, TN, October 19, 2008.
115. “Object-Oriented Patterns for Concurrent and Networked Applications,” Qualcomm, San Diego, CA, August 5-6th, 2008.
116. “The C++ Standard Template Library,” Qualcomm, San Diego, NJ, July 29-30, 2008.

117. "Object-Oriented Patterns and Frameworks with C++," Qualcomm, San Diego, CA, June 12-13, 2008.
118. "The C++ Standard Template Library," Qualcomm, New Jersey, May 5-6, 2008.
119. "Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing," Software Architecture Technology Users' Network (SATURN) workshop April 28 - May 1, 2008 in Pittsburgh, PA.
120. Developing Distributed Computing Systems with Patterns and Middleware, UCLA Extension, February 19-21, 2008.
121. Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing, OOPSLA 2007, Montreal, CA, October 24, 2007.
122. Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware, Qualcomm, New Jersey, September 27-28, 2007.
123. Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware, Qualcomm, San Diego, CA, August 21-22, 2007.
124. Lightweight CORBA Component Model, 8th OMG Real-time/Embedded CORBA workshop, Washington DC, July 9–12, 2007.
125. Model-Driven Engineering for Distributed Real-time and Embedded Systems, 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2007), Bellevue, WA, United States April 3-6, 2007.
126. "Improving Product Reliability and ROI Through Effective Software Reuse," Qualcomm, San Diego, CA, March 27th, 2007.
127. "Developing Distributed Computing Systems with Patterns and Middleware," UCLA Extension, February 21-23, 2007.
128. "POSA: Patterns for Concurrent and Distributed Systems," OOP, Munich, Germany, January 22, 2007.
129. "Meeting the Challenges of Software-Intensive Embedded Systems," OOP, Munich, Germany, January 23, 2007.
130. "Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware," Qualcomm, San Diego, CA, January 10-11, 2007.
131. "Model-Driven Development of Distributed Systems," OOPSLA 2006, Portland, OR, October 22-26, 2006.
132. "Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects," OOPSLA 2006, Portland, OR, October 22-26, 2006.
133. "Model-Driven Engineering of Distributed Systems," MODELS 2006, Genova, Italy, October 1, 2006.
134. "Distributed Real-time and Embedded Systems," Advanced Institute of Information Technology, Seoul, Korea, August 7-11 2006.
135. "Lightweight CORBA Component Model," 7th OMG Real-time/Embedded CORBA workshop, Washington DC, July 10–13, 2006.
136. "How to Use ACE Effectively," Trion World Network, Austin, TX, June 19-21, 2006.
137. "Improving Product Reliability and ROI Through Effective Software Reuse," Qualcomm, San Diego, CA, June 15, 2006.
138. "Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware," Qualcomm, San Diego, CA, June 13-14, 2006.
139. "Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware," Qualcomm, San Diego, CA, Feb 9-10, 2006.
140. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, University of California, Los Angeles Extension, January 18-20st, 2006."
141. "Model Driven Development of Distributed Real-time and Embedded Systems," at the OOP conference, January 17, 2006, Munich, Germany.

142. "Pattern-Oriented Software Architecture," at the OOP conference, January 16, 2006, Munich, Germany.
143. "Model Driven Development: State of the Art," at the OOP conference, January 16, 2006, Munich, Germany.
144. "Concurrent C++ Network Programming with Patterns and Frameworks," C++ Connections: 20 Years of C++ conference, November 11, 2005, Mandalay Bay, Las Vegas, NV.
145. "Pattern-Oriented Software Architecture: Patterns for Concurrent and Distributed Systems," OOPSLA 2005, San Diego, October 17th, 2005.
146. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," BAE Systems, Greenlawn, New York, August 25, September 2-3.
147. "Lightweight CORBA Component Model," 6th OMG Real-time/Embedded CORBA workshop, Washington DC, July 11–14, 2005.
148. "Model Driven Development for Distributed Real-time and Embedded Systems," OMG Information Days: MDA - Frankfurt, Germany, June 9th, 2005
149. "Model Driven Development for Distributed Real-time and Embedded Systems," OMG Information Days: MDA - Munich, Germany, June 7th, 2005.
150. "Model Driven Development for Distributed Real-time and Embedded Systems," OMG Information Days: MDA - Zurich, Switzerland, June 1st, 2005.
151. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," BAE Systems, Wayne, New Jersey, May 13, 16, 19, 23, 27, 2005.
152. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," BAE Systems, Wayne, New Jersey, February 18th, February 22nd, March 1, 8, and 15 2005.
153. "Pattern-Oriented Software Architectures for Distributed Systems" the OOP conference, January 28, 2005, Munich, Germany.
154. "Research on Model Driven Development of Distributed Real-time and Embedded Systems," the OOP conference, January 26, 2005, Munich, Germany.
155. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, January 19-21st, 2005.
156. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, BAE Systems, Wayne, New Jersey, October 29, November 1, 8, 15, 22, 2004.
157. "Pattern-Oriented Software Architectures for Distributed Systems," OOPSLA 2004, Vancouver, British Columbia, October 25th, 2004.
158. "Notes on the Forgotten Craft of Software Architecture", OOPSLA 2004, Vancouver, British Columbia, October 25th, 2004.
159. "Model Driven Architecture with QoS-enabled component middleware," MDE for Embedded Systems, Brest, France, September 10th 2004.
160. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Qualcomm, San Diego, CA, Jan 7-6, 2005.
161. "Object-Oriented Design and Programming with Patterns, Frameworks, and Middleware," Qualcomm, San Diego, CA, Jan 9-10, 2005.
162. "Using the Lightweight CORBA Component Model to Develop Distributed Real-time and Embedded Applications," OMG Workshop on Distributed Object Computing for Real-time and Embedded Systems, July 12th, 2004, Reston, VA.
163. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, July 7-9th, 2004.
164. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, University of California, Los Angeles Extension, January 21st-23rd, 2004.
165. Patterns and Frameworks for Concurrent Distributed Systems, SIGS OOP Conference, Munich, Germany, January 19th, 2004.

166. Middleware for Distributed Real-time and Embedded Systems, SIGS OOP Conference, Munich, Germany, January 19th, 2004.
167. "Pattern-Oriented Software Architectures for Networked and Concurrent Applications," OOPSLA 2003, Anaheim, CA, October 27, 2003.
168. The JAOO 2003 conference, September 22-26, Aarhus, Denmark.
169. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, July 9-11th, 2003.
170. "Patterns, Frameworks, and Middleware: Their Synergistic Relationship," Frontiers of Software Practice, International Conference on Software Engineering, Portland, Oregon, May 7, 2003.
171. "Pattern-Oriented Distributed Systems Architecture," International Conference on Software Engineering, Portland, Oregon, May 5, 2003.
172. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, January 22nd-24th, 2003.
173. "Patterns and Application Experiences for Real-time Object Request Brokers," OOPSLA 2002, Seattle, Washington, November, 2002.
174. "Pattern-Oriented Software Architectures for Networked and Concurrent Applications," OOPSLA 2002, Seattle, Washington, November, 2002.
175. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, Raytheon, St. Petersburg, FL, September 3–5, 2003.
176. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, University of California, Los Angeles Extension, July 22nd-24th, 2002.
177. "Policies and Patterns for High-performance, Real-time Object Request Brokers," Mercury Computer Systems, Tysons Corner, VA, November Feb 7, 2002.
178. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, University of California, Los Angeles Extension, January 23rd-25th, 2002.
179. "Policies and Patterns for High-performance, Real-time Object Request Brokers," Raytheon, Rosslyn, VA, November 12th, 2001.
180. "Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects," OOPSLA 2001, October 15th, 2000, Minneapolis, Minnesota.
181. "Policies and Patterns for High-performance, Real-time Object Request Brokers," International Symposium on Distributed Object Applications (DOA), Rome, September 17-20, 2001.
182. "Policies and Patterns for QoS-enabled Middleware," The JAOO 2001 conference, September 10-14, Aarhus, Denmark.
183. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, July 23rd-25th, 2001.
184. "Policies and Patterns for High-performance, Real-time Object Request Brokers," OMG Second Workshop on Real-time and Embedded Distributed Object Computing on June 4-7, 2001 in Herndon, VA, USA.
185. "Design Patterns for Understanding Middleware and Component Infrastructures," 6th USENIX Conference on Object-Oriented Technologies and Systems, January 29, 2001, San Antonio, TX.
186. "Principles and Patterns of High-performance, Real-time Object Request Brokers," OOP conference, Munich, Germany, January 23, 2001.
187. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, January 3-5, 2001.
188. "Patterns for Concurrent and Distributed Objects," OOPSLA 2000, October 16th, 2000, Minneapolis, Minnesota.
189. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Berkeley Extension, May 24-26, 2000.
190. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Jet Propulsion Laboratory, Pasadena, CA, April, 2000.

191. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Los Angeles Extension, March 27-31, 2000.
192. "Optimizing Middleware to Support High-Performance Real-time Distributed and Embedded Systems," OOP conference, Munich, Germany, January 27, 2000.
193. "Effective Architectures for DOC," OOP conference, Munich, Germany, January 24, 2000.
194. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California, Berkeley Extension, December 13-15, 1999.
195. "Middleware Techniques and Optimizations for Real-time Embedded Systems," 12th International Symposium On System Synthesis, IEEE, San Jose, CA, USA November, 11, 1999
196. "Patterns and Principles of Real-time Object Request Brokers," OOPSLA 1999, ACM, Denver, Colorado, November 1-5, 1999.
197. "Using Design Patterns, Frameworks and CORBA to Reduce the Complexity of Developing Reusable Large-Scale Object-Oriented Concurrent Communication Components and Systems," Fifth IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, Nevada, October 18-21, 1999
198. "Distributed Technologies," Motorola, Schaumburg, IL, August 10-12, 1999.
199. "Patterns and Principles of Real-time Object Request Brokers," the 3rd Components Users Conference, SIEMENS, Munich, Germany, July 12th, 1999.
200. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent, Naperville, IL, June 23-24 and June 30 - July 1st, 1999.
201. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Motorola Software Symposium, Ft. Lauderdale, Florida, June 21st, 1999.
202. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California Los Angeles Extension, June 2-4, 1999.
203. "Concurrent Object-Oriented Network Programming and Distributed Object Computing," University of California Berkeley Extension, May 19-21, 1999.
204. "Patterns and Principles of Real-time Object Request Brokers," 5th USENIX Conference on Object-Oriented Technologies and Systems, May 4, 1999, San Diego, CA.
205. "Real-time CORBA for Telecom – Fact or Fiction?" Nortel Design Forum, Ottawa, CA, April 22, 1999.
206. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent, Columbus, OH, March 18-19 and 25-26, 1999.
207. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent, Holmdel, NJ, March 1-4, 1999.
208. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent/Octel, Milpitas, CA, December 14-16, 1998.
209. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California Los Angeles Extension, December 8-10, 1998.
210. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Motorola, Schaumburg, IL, December 2-4, 1998.
211. "Concurrent Object-Oriented Network Programming and Distributed Object Computing," University of California Berkeley Extension, November 16-18, 1998.
212. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Software," OOPSLA 1998, October 19th, 1998, Vancouver, British Columbia.
213. "High-Performance CORBA," Lucent CORBA Forum, Holmdel, NJ, September 29, 1998.
214. "Writing Efficient Multi-Thread CORBA Applications," the 3rd Components Users Conference, SIEMENS, Munich, Germany, July 10, 1998.
215. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Software," UCLA extension course, Milan, Italy, June 29 - July 1, 1998.

216. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," Lucent, Naperville, IL, June 8-11, 1998.
217. "Patterns and Performance of Real-time Object Request Brokers," Fourth IEEE Real-Time Technology and Applications Symposium (RTAS), Denver, Colorado, June 5, 1998.
218. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California Los Angeles Extension, June 1-3, 1998.
219. "Patterns and Principles of Real-time Object Request Brokers," NSA, Ft. Meade, MD, March 22, 1998.
220. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, Crosskeys, Ottawa Canada, March 19-21, 1998.
221. "Concurrent Object-Oriented Network Programming and Distributed Object Computing," University of California Berkeley Extension, March 4-6, 1998.
222. "Building Distributed Communication Software with CORBA," the Motorola Systems Symposium, February, 1998, Austin, Texas, USA.
223. "Introduction to Distributed Objects with CORBA," SIGS OOP '98, February 9-13, 1998, Munich, Germany.
224. "Design Patterns for Developing and Using CORBA Object Request Brokers," SIGS OOP '98, February 9-13, 1998, Munich, Germany.
225. Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems, Lucent Technologies, Whippany, NJ, January 5-6, 1998.
226. "Using Design Patterns, Frameworks, and CORBA to Develop Object-Oriented Communication Systems," University of California Los Angeles Extension, December 10-12, 1997.
227. "Concurrent Object-Oriented Network Programming and Distributed Object Computing," University of California Berkeley Extension, December 10-12, 1997.
228. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Systems," Motorola Cellular Infrastructure Group, Arlington Heights, Illinois, December 1 - 3, 1997.
229. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Systems," TOOLS Pacific '97, Melbourne, Australia November 24 - 27, 1997.
230. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Systems" for the IEEE GLOBECOM '97 conference, Phoenix, AZ, November 4-8, 1997.
231. "High-performance Distributed Object Computing with CORBA," IEEE International Conference on Network Protocols, Atlanta, GA, October 28th, 1997.
232. "Using Design Patterns and Frameworks to Develop Object-Oriented Communication Systems," OOPSLA 1997, ACM, Atlanta, GA, October 6-7th, 1997.
233. "Using Design Patterns and Frameworks to Develop Object-oriented Communication Systems," 24th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS Asia '97). Beijing, China, September 22, 1997.
234. "Principles and Patterns of Distributed Object Computing Systems," for the ACM Principles of Distributed Computing Conference (PODC), Santa Barbara, CA, August 21st, 1997.
235. "Distributed Object Computing with CORBA and ACE," Alta Software, Jacksonville, FL, June 4-5th, 1997.
236. "Distributed Object Computing with CORBA", Object Expo, NY, NY, June 2nd, 1997.
237. "Concurrent Object-Oriented Network Programming and Distributed Object Computing," University of California Berkeley Extension, May 28-30, 1997.
238. "Patterns and Principles of Real-time Object Request Brokers," National Security Agency, Ft. Meade, MD, May 13th, 1997.
239. "Building Distributed Communication Software with CORBA," the Motorola Systems Symposium, March, 1997, Chandler, AZ, USA.
240. "Evaluating Concurrency Models for CORBA Servers," the 2nd Components Users Conference, SIEMENS, Munich, Germany, July 14th, 1997.

241. "Design Patterns for Evolving System Software Components from UNIX to Windows NT," the 2st Components Users Conference, SIEMENS, Munich, Germany, July 14th, 1997.
242. "Techniques and Patterns for Distributed Object Computing with CORBA and C++," University of California Berkeley Extension, December 4-6, 1996.
243. "Design Patterns for Concurrent Object-Oriented Programming with ACE and C++," C++ World, Dallas, TX, November 11th, 1996.
244. "Implementing Concurrent CORBA Applications with Multi-Threaded Orbix and ACE," C++ World, Dallas, TX, November 12th, 1996.
245. "Why Reuse has Failed, and How You Can Make it Work for You," Berne Technology Forum 1996, Berne, Switzerland, October 18, 1996.
246. "Introduction to Distributed Object Programming with CORBA," the Local Computer Networks '96 conference, IEEE, Minneapolis, Minnesota, October 13, 1996.
247. "Object-Oriented Design Patterns for Concurrent, Parallel, and Distributed Systems," the OOPSLA 1996 conference, ACM, San Jose, California, October, 1996.
248. "OO Design Patterns Network Programming in C++," Object Expo Europe, London, England, September 23rd, 1996.
249. "Effective Multithreaded CORBA Programming," Object Expo Europe, London, England, September 24th, 1996.
250. "Workshop on Object Oriented Technologies," Mitsubishi, July 22nd to July 26th, 1996, Kobe, Japan.
251. "Evaluating Concurrency Models for CORBA Servers," the 1st Components Users Conference, SIEMENS, Munich, Germany, July 15th, 1996.
252. "Design Patterns for Evolving System Software Components from UNIX to Windows NT," the 1st Components Users Conference, SIEMENS, Munich, Germany, July 15th, 1996.
253. "OO Design Patterns for Concurrent, Parallel, and Distributed Systems," the $2^{nd}$ *Conference on Object-Oriented Technology*, USENIX, Toronto, Canada, June 17, 1996.
254. "OO Design Patterns for Concurrent, Parallel, and Distributed Systems," the $3^{rd}$ *Conference on Object-Oriented Technology*, USENIX, Portland, Oregon, June 16th, 1996.
255. "OO Design Patterns for Network Programming in C++," the *Object Expo '96 Conference*, SIGS, Sydney, Australia, June $3^{rd}$, 1996.
256. "Effective Multi-threaded CORBA Programming Programming," the *Object Expo '96 Conference*, SIGS, Sydney, Australia, June $5^{th}$, 1996.
257. "Concurrent Object-oriented Network Programming with C++," University Of California Berkeley Extension, Berkeley, California, May $22^{nd} - -24^{th}$, 1996.
258. "Experience Developing Reusable Software Using Object-Oriented Design Patterns and Frameworks," the $4^{th}$ *International Conference on Software Reuse*, Orlando, Florida, USA April 23-26, 1996.
259. "Techniques for Object-Oriented Network Programming," the *OOP Conference*, SIGS, Munich, Germany, Feb 14th, 1996.
260. "Using Object-Oriented Design Patterns to Develop Large-Scale Distributed Systems," the *OOP Conference*, SIGS, Munich, Germany, Feb $13^{th}$, 1996.
261. "Concurrent Object-oriented Network Programming with C++," University Of California Berkeley Extension, Berkeley, California, November 30th-December 1st, 1995.
262. "Using Object-Oriented Design Patterns to Develop Large-Scale Distributed Systems," the $4^{th}$ *C++ World Conference*, SIGS, Chicago, Illinois, October 31st, 1995.
263. "Techniques for Object-Oriented Network Programming," the $4^{th}$ *C++ World Conference*, SIGS, Chicago, Illinois, October 31st, 1995.
264. "Experience using OO Design Patterns to Develop Large-scale Distributed Communication Systems," *OOPSLA 1995 Conference* in Austin, Texas, October 1995.

265. "Concurrent Object-oriented Network Programming with C++," the $9^{th}$ *European Conference on Object-Oriented Programming (ECOOP)*, Aarhus, Denmark, August, 1995.
266. "Concurrent Object-Oriented Network Programming with C++," the $1^{st}$ *Conference on Object-Oriented Technology*, USENIX, Monterey, California, June 23, 1995.
267. "Design Patterns for Concurrent and Distributed Systems," the *Object Expo '95 Conference*, SIGS, New York, NY, June $5^{th}$ 1995.
268. "Object Oriented Network Programming," the *Object Expo '95 Conference*, SIGS, New York, NY, June $5^{th}$, 1995.
269. "Software Construction with Active Objects in C++," the *OOP '95 Conference*, SIGS, Munich, Germany January 31, 1995.
270. "Object-Oriented Concurrent Programming with C++," the *OOP '95 Conference*, SIGS, Munich, Germany January 31, 1995.
271. "Concurrent Object-Oriented Programming," the *Winter USENIX Conference*, USENIX, New Orleans, Louisiana, January, 1995.
272. "Object-Oriented Network Programming with C++," the $3^{rd}$ *C++ World Conference*, SIGS, Austin, Texas, November 14, 1994.
273. "Object-Oriented Techniques for Dynamically Configuring Concurrent Distributed Applications," the $9^{th}$ *OOPSLA 1994*, ACM, Portland, Oregon, October 23, 1994.
274. "Object-Oriented Network Programming," the $6^{th}$ *C++ Conference*, USENIX, Cambridge, Massachusetts, April 11, 1994.
275. "Object-Oriented Techniques for Developing Extensible Network Servers," the $2^{nd}$ *C++ World Conference*, SIGS, Dallas, Texas, October 19, 1993.

# Professional Activities

### Editorial Activities

1. Guest co-editor for a special issue of the Springer Journal Annals of Telecommunications on "Middleware for Internet distribution in the context of Cloud Computing and the Internet of Things," 2016, with Gordon Blair and Chantal Taconet.
2. Guest co-editor of the Proceedings of the IEEE special issue on Applications of Augmented Reality Environments, 2014.
3. Guest co-editor of the International Journal of Network Protocols and Algorithms (NPA) Special Issue on Data Dissemination for Large scale Complex Critical Infrastructures, 2010.
4. Wrote the foreword to the book *Patterns of Parallel Software Design* by Jorge Luis Ortega Arjona, Wiley, 2010.
5. Editorial board member of the Springer Journal of Internet Services and Applications (JISA).
6. Editorial board member of the Transactions on Pattern Languages of Programming (TPLoP) published by Springer-Verlag.
7. Wrote the foreword to the book *Practical Software Factories in .NET*, by Gunther Lenz and Christoph Wienands, Apress, 2006.
8. Guest editor of the IEEE Computer Special Issue on Model Driven Development, February 2006.
9. Guest co-editor of IEEE Network special issue on "Middleware Technologies for Future Communication Networks," February 2004 (co-editors with Gordon Blair and Andrew Campbell).
10. Editorial board member of the Springer Journal of Aspect-Oriented Software Development.
11. Wrote the foreword to the book *Fundamentals of Distributed Object Systems: The CORBA Perspective*, by Zahir Tari and Omran Bukhres, Wiley and Sons, 2001.
12. Wrote the foreword to the book *Design Patterns in Communication Software*, edited by Linda Rising, Cambridge University Press, 2000.
13. Guest editor of the Special Issue on Components and Patterns for *The Journal of Theory and Practice of Object Systems*, Wiley & Sons, to appear 2002.

14. Invited editorial on "Trends in Distributed Object Computing" for the special issue on Distributed Object-Oriented Systems appearing in the Parallel and Distributed Computing Practices journal, edited by Maria Cobb and Kevine Shaw, Vol. 3, No. 1, March 2000.
15. Co-editor of "Building Application Frameworks: Object-Oriented Foundations of Framework Design," John Wiley & Sons, 1999 (co-editors are Mohamed Fayad and Ralph Johnson), ISBN 0-471-24875-4.
16. Co-editor of "Implementing Application Frameworks: Object-Oriented Frameworks at Work," John Wiley & Sons, 1999 (co-editors are Mohamed Fayad and Ralph Johnson), ISBN 0-471-25201-8.
17. Guest editor of the Special Issue on OO Application Frameworks for the Communications of the ACM, (co-editor Mohamed Fayad), ACM, October, 1997.
18. Guest editor of the special issue on Distributed Object Computing for USENIX Computing Systems Journal, November/December, 1996.
19. Guest editor of a feature topic on Distributed Object Computing for IEEE Communications Magazine, February, 1997.
20. Wrote the foreward for Dr. Nayeem Islam's book on *Distributed Objects: Methodologies for Customizing Operating Systems* (IEEE Computer Society Press, 1996).
21. Guest editor of the Special Issue on Patterns and Pattern Languages for Communications of the ACM, (co-editors Ralph Johnson and Mohamed Fayad), ACM, October, 1996.
22. Co-editor of a book entitled "Pattern Languages of Program Design," Addison-Wesley, 1995 (co-editor is Jim Coplien, Bell Labs).
23. Editor of the Patterns++ section of the C++ Report Magazine, April 1997 - March 1998.
24. Editor-in-chief of the C++ Report Magazine, January 1996 - February 1997.
25. Editorial board member of the IEEE Computer Society - Computer Science & Engineering Practice Board.

**Program Chairmenships and Conference Organization**

1. Chair of the DoD Organic Software Infrastructure Workshop, Arlington VA, August 13th, 2018.
2. General Chair of the Software Product Line Conference, Nashville TN, July/August, 2015.
3. Program Chair of the Interoperable Open Architecture 2013 conference, September 10-11, 2013, Washington, DC.
4. Program Chair of the NSF Workshop on Computing Clouds for Cyber-Physical Systems, March 15th, 2013, Ballston, VA.
5. Program Chair of the Interoperable Open Architecture 2012 conference, October 29-31, 2012, London, UK.
6. Program co-chair for the 1st International Symposium on Secure Virtual Infrastructures (DOA-SVI'11), 17-19 Oct 2011, Crete, Greece.
7. Program co-chair for the COMmunication System softWAre and middleware (Comsware) conference, Helsinki, Finland, August 2010.
8. Doctoral symposium chair for OOPSLA 2009, Orlando Floria, October 25-29, 2009.
9. General co-chair for the 3rd ACM International Conference on Distributed Event-Based Systems (DEBS 2009), July 6-9, 2009 - Nashville, TN, USA.
10. Member of the ISORC 2009 advisory and publicity committee for ISORC 2009, March 17-20, 2009, Toyko, Japan.
11. Area Coordinator for the Integrating Systems of Systems using Services topic at the 6th International Conference on Service Oriented Computing, Sydney (Australia), December 1st - 5th, 2008.
12. Member of the Advisory and Publicity Committee for ISORC 2008, Orlando, Florida, May 5 -7, 2008.
13. Co-chair of the Middleware for Network Eccentric and Mobile Applications (MiNEMA.08) Workshop co-located with ACM EuroSys Conference, March 31 - April 1, 2008, Glasgow, Scotland.

14. General chair of the ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (MODELS 2007), Nashville TN, September 30-October 5, 2007.
15. Area co-coordinator for the Quality of Service research track at the The Fifth International Conference on Service-Oriented Computing, September 17-20, 2007, Vienna, Austria.
16. Program co-chair of the NSF workshop on New Research Directions in Composition and Systems Technology for High Confidence Cyber Physical Systems, July 9, 2007.
17. Program co-chair for the Science of Design Principal Investigators workshop, February 28 to March 2, 2007.
18. Program co-coordinator for SOA Runtime area of the 4th International Conference on Service Oriented Computing Chicago, USA, December 4-7, 2006.
19. Program co-chair of the NSF/NCO Workshop on High-Confidence Software Platforms for Cyber-Physical Systems (HCSP-CPS) Workshop systems, November 30th to December 1st, 2006, Alexandria, VA.
20. Panels chair for the MoDELS 2006 conference, Genova Italy, Oct. 2-6, 2006.
21. Program Co-Chair of the Generative Programming and Component Engineering (GPCE) Conference, Portland, OR, October 2006 (collocated with OOPSLA '06).
22. Program Chair of the NSF/NCO Workshop on New Research Directions in High Confidence Software Infrastructure for Distributed Real-time and Embedded (DRE) systems, July 10th, 2006, Fairfax VA.
23. Program Co-Chair of the NSF/NCO High Confidence Medical Device Software and Systems (HCMDSS) Workshop, May 2005, University of Pennsylvania, Philadelphia, Pennsylvania.
24. Track Vice Chair for Real-time Middleware and Software Engineering for the Real-time Systems Symposium, Lisbon, Portugal, December, 2004.
25. Program Co-chair for the NSF/NCO Planning Meeting for the High Confidence Medical Device Software and Systems (HCMDSS) Workshop, November 16-17, 2004, Arlington, VA.
26. Program chair for 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOSPLA), October 24-28, 2004, Vancouver, British Columbia, Canada.
27. General co-chair of the IEEE Real-Time and Embedded Technology and Applications Symposium, May 25 – 28, 2004, Toronto, Canada.
28. Program chair of the CCM Workshop, December 10th, 2003, Nashville, TN.
29. General co-chair for the 5th International Symposium on Distributed Objects and Applications, November 3–7 2003, Catania, Sicily.
30. Program co-chair of the 3rd TAO Workshop, July 18, 2002, Arlington, VA.
31. Program co-chair for Middleware 2003, 4th IFIP/ACM/USENIX International Conference on Distributed Systems Platforms, June 16-20, 2003, Rio de Janeiro, Brazil.
32. Program co-chair for the 9th IEEE Real-time/Embedded Technology and Applications Symposium (RTAS), May 27-30, 2003, Washington, DC.
33. Area vice-chair and session chair for Middleware at the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), May 19-22nd, 2003, Providence, RI.
34. Program co-chair of the IEEE Workshop on LargeScale Real-Time and Embedded Systems, December 2, 2002, Austin, TX.
35. Program co-chair for the 4th International Symposium on Distributed Objects and Applications, October 28–November 1, 2002, Irvine, CA.
36. Co-organizer of the cross-agency Software Design and Productivity Coordinating Group Workshop on New Visions for Software Design and Productivity: Research and Applications, December 13-14, Nashville, TN.
37. Program co-chair for the 3rd International Symposium on Distributed Objects and Applications, September 18-20, 2001, Rome, Italy.

38. Co-organizer of the cross-agency Workshop on New Visions for Software Design and Productivity, April 18-19, 2000, Ballston, VA.
39. Area vice-chair and session chair for Middleware at the IEEE International Conference on Distributed Computing Systems, April 16-19, Phoenix, AZ, 2001.
40. Tutorial chair for the 6th USENIX Conference on Object-Oriented Technologies and Systems, January 27 - February 3, 2001, San Antonio, TX.
41. Co-chair of the OMG Workshop on Real-time and Embedded CORBA, in Reston, VA, July 24-27, 2000.
42. General chair of the IFIP/ACM International Conference Middleware 2000 in New York, April, 2000.
43. Tutorial chair for the $5^{th}$ USENIX Conference on Object-Oriented Technologies and Systems, May 3-7, 1999, San Diego, CA.
44. Treasurer for the Fourth International Workshop on Object-oriented Real-time Dependable Systems (WORDS'99) January 27-29, 1999, Radisson Hotel, Santa Barbara, California, USA.
45. Tutorial chair for the $4^{th}$ USENIX Conference on Object-Oriented Technologies and Systems, April 27-30, 1998, Santa Fe, New Mexico.
46. Co-chair of the mini-track on Engineering Client-Server Systems for the HICSS-31 conference, the Big Island of Hawaii - January 6-9, 1998.
47. Tutorial chair for the $3^{rd}$ USENIX Conference on Object-Oriented Technologies and Systems, Portland, OR, June 1997.
48. Publicity chair for the $5^{th}$ IEEE International Workshop on Object-Orientation in Operating Systems, IEEE TCOS and USENIX, Seattle, Washington, October 27-28, 1996.
49. Program chair for $3^{rd}$ conference on Programming Languages of Programming, Allerton, IL, USA, September, 1996.
50. Program chair for the $2^{nd}$ USENIX Conference on Object-Oriented Technologies, June 1996.

**Professional Service and Advisory Positions**

1. Member of the Fraunhofer Advisory Board for the University of Maryland, College Park.
2. Member of the steering committee for the Software Product-Line Conference series.
3. Member of the Future Airborne Capabilities Environment (FACE) Advisory Board.
4. Vice-Chair of the Cyber Situation Awareness study for the Air Force Scientific Advisory Board.
5. Member of the Joint Tactical Radio System (JTRS) Tiger Team in support of the Assistant Secretary of the Army, Acquisition, Logistics, and Technology.
6. Member of the Air Force Scientific Advisory Board.
7. Member of the advisory board for the NSF-sponsored Repository for Model-Driven Development (ReMoDD) project at Colorado State University.
8. Member of the National Academics Committee on Advancing Software-Intensive Systems Producibility, chaired by Bill Scherlis from Carnegie Mellon University (CMU).
9. Member of the Engineering and Methods Technical Advisory Group (TAG) for the Software Engineering Institute at Carnegie Mellon University (CMU) from 2006 to 2009.
10. Member of the Ultra-Large-Scale (ULS) Systems study commissioned by the US Army and conducted at the Software Engineering Institute at Carnegie Mellon University (CMU).
11. Member of the Joshua group, which is an advisory board for the Air Force Research Lab (AFRL) in Rome, NY.
12. Member of the steering committee for the Distributed Objects and Applications conference series.
13. Member of the steering committee for the ACM/USENIX/IFIP Middleware conference series.
14. Member of the steering committee for EMSOFT 2002: Second Workshop on Embedded Software, Grenoble, France, October, 7–9th, 2002.

15. Member of the steering committee for EMSOFT 2001: First Workshop on Embedded Software, Lake Tahoe, California, October, 8th–10th, 2001.
16. Member of the Board of Directors for the Embedded Systems Consortium for Hybrid and Embedded Research (ESCHER).
17. Member of the NASA/JPL Mars Science Laboratory Mission Concept Review Board.
18. Chair of the subcommittee on Embedded and Hybrid Systems program for the National Science Foundation's 2003 Committee of Visitors in the Computer and Communications Research (C-CR) Division.
19. Co-chair of the Software Design and Productivity (SDP) Coordinating Group of the Federal government's multi-agency Information Technology Research and Development (IT R&D) Program, the collaborative IT research effort of the major Federal science and technology agencies. The SDP Coordinating Group formulates the multi-agency research agenda in fundamental software design.
20. One of the three founding members of the Scientific Advisory Board for the *International Symposium of Distributed Objects and Applications* conference series.
21. Member of the advisory board for Entera, which provides Internet content delivery systems based on ACE.
22. Invited to participate in the OO Working Group of the "Strategic Directions in Computing Research" workshop sponsored by ACM at MIT in June 1996.

**Technical Program Committees**

1. Middleware 2021 Doctoral Symposium, Dec. 6-10, 2021 in Quebec Canada.
2. The 2nd IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS 2021), September 27 to October 1, 2021, Washington DC, USA.
3. "Web of Things, Ubiquitous and Mobile Computing" Track for the Web Conference 2021, Ljubljana, Slovenia, from April 19-23, 2021.
4. 7th International Workshop on Middleware and Applications for the Internet of Things (M4IoT), December 2020 in conjunction with the ACM/IFIP International Middleware Conference.
5. 14th ACM International Conference on Distributed and Event-based Systems, July 13 to July 17, 2020, in Montreal, Quebec, Canada.
6. The Web Conference 2020: Web of Things, Ubiquitous, and Mobile Computing Track, April 20-24th, 2020, Taipei, Taiwan.
7. 6th Middleware for Context-Aware Applications in the IoT (M4IOT) workshop collocated with the ACM/IFIP/USENIX Middleware 2019 Conference, UC Davis, California, USA, December 9-13th 2019.
8. IEEE Workshop on IoT Big Data and Blockchain, at the 2019 IEEE International Conference on Big Data (IEEE Big Data 2019), December 9-12, 2019, Los Angeles, CA, USA.
9. The Second International Workshop on Blockchain Dependability, in conjunction with SRDS2019, Lyon, France, October 1, 2019.
10. The 13th ACM International Conference on Distributed and Event-based Systems, 4th-28th June, 2019, Darmstadt, Germany.
11. The "Web of Things, Ubiquitous, and Mobile Computing" track of The Web Conference 2019, San Francisco, CA, USA, May 13–17, 2019.
12. 17th Workshop on Adaptive and Reflexive Middleware (ARM), collocated with ACM/IFIP/Usenix Middleware 2018, December 10-14th, 2018, Rennes, France.
13. 25th International Conference on Pattern Languages of Programs (PLoP 2018), October 23 – 26th, Portland, OR, USA.
14. First International Workshop on Blockchain Dependability (WBD2018), held in conjunction with the 14th European Dependable Computing Conference, 10-14 September 2018, Iasi, Romania.
15. Workshop on Designing Resilient Intelligent Systems for Testability and Reliability, April 30 – May 4, 2018 in Seattle, USA (co-located with ICSA 2018).

16. 15th IEEE International Conference on Autonomic Computing (ICAC 2018), Sept 3-7, 2018, Trento, Italy.
17. International Conference on Information Society and Smart Cities (ISC 2018), Oxford city, United Kingdom 06-07 June, 2018.
18. 16th Workshop on Adaptive and Reflective Middleware workshop collocated with the ACM/IFIP/USENIX Middleware 2017 Conference, Las Vegas, Nevada, Dec 11-15, 2017.
19. 4th Middleware for Context-Aware Applications in the IoT (M4IOT) workshop collocated with the ACM/IFIP/USENIX Middleware 2017 Conference, Las Vegas, Nevada, Dec 11-15, 2017.
20. 10th International Workshop on Dynamic Software Product Lines - Adaptive Systems through Runtime Variability (DSPL '17), Sept 25-29, 2017, Sevilla, Spain.
21. 11th ACM International Conference on Distributed and Event-Based Systems (DEBS 2017), June 19 - 23, 2017, Barcelona, Spain.
22. 3rd Middleware for Context-Aware Applications in the IoT (M4IOT) workshop collocated with the ACM/IFIP/USENIX Middleware 2016 Conference, December 12-16, 2016 - Trento, Italy.
23. 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, October 5th – 14th, 2016, Corfu, Greece.
24. 10th ACM International Conference on Distributed and Event-based Systems, June 20 to June 24, 2016 in Irvine, CA.
25. First International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE), April 11, 2016, Vienna, Austria (Co-located with CPS Week).
26. 9th Dynamic Software Product Lines (DSLP) 2015 (held as part of SASO 2015) at MIT on September 21, 2015.
27. 13th International Conference on Advances in Mobile Computing and Multimedia (MoMM2015), Brussels, Belgium from 10-12 December 2015.
28. 13th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2015, track on Cyber Physical Systems, Porto Portugal, October 21-23, 2015.
29. 35th IEEE International Conference on Distributed Computing Systems (ICDCS), June29 - July 2, 2015 in Columbus, Ohio, USA.
30. Fourth International Conference on Emerging Applications of Information Technology (EAIT) at Indian Statistical Institute, Kolkata, India, December 19-21, 2014.
31. The 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2014), Berlin, Germany, April 2014.
32. International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2014), Lisbon, Portugal, 7-9 January, 2014.
33. 14th ACM/IFIP/USENIX International Middleware Conference (Middleware 2013), December 9-13, Beijing, China.
34. 32nd International Symposium on Reliable Distributed Systems (SRDS 2013), September 30-October 3, 2013 at Braga, Portugal.
35. 17th International Software Product Line Conference SPLC, Tokyo, Japan, 26-30 August 2013.
36. First International Workshop on Engineering Mobile-Enabled Systems, in conjunction with ICSE 2013, May 18-26th, 2013, San Francisco, CA.
37. International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2013), Barcelona, Spain, 19-21 February, 2013.
38. ACM/USENIX/IFIP International Middleware conference, Montreal, Quebec, Canada, December 3-7, 2012.
39. 11th Workshop on Adaptive and Reflective Middleware, in conjunction with Middleware 2012 in Montreal, Quebec, Canada, December 3-7, 2012.
40. International Workshop on Real-Time and Distributed Computing in Emerging Applications (REACTION) 2012, San Juan, Puerto Rico, December 4, 2012, in co-location with the 33rd IEEE Real-Time Systems Symposium.

41. Third International Conference on Emerging Applications of Information Technology (EAIT) November 29 - December 01, 2012, Kolkata, India.
42. IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS), Las Vegas, USA, November 12 - 14, 2012.
43. 31st International Symposium on Reliable Distributed Systems (SRDS), 8th-11th October 2012. Irvine, California.
44. Sixth International Workshop on Dynamic Software Product Lines (DSPL), September 2 - 7, 2012, Salvador, Brazil.
45. 16th International Software Product Line Conference (SPLC 2012), Salvador, Brazil on 02-07 September 2012.
46. 5th International workshop UML and Formal Methods (UML&FM 2012), Paris, France, August 27-31, 2012.
47. UML&AADL 2012, July 18-20,2012, Ecole Normale Superieure, Paris, France.
48. 17th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2012), July 18-20,2012, Ecole Normale Superieure, Paris, France.
49. COMPSAC 2012 - Trustworthy Software Systems for the Digital Society, July 16-20, 2012, Izmir, Turkey.
50. Foundations Track of the 8th European Conference on Modelling Foundations and Applications (ECMFA 2012), Copenhagen, Denmark, 2-6th of July, 2012.
51. 24th International Conference on Software Engineering and Knowledge Engineering, Redwood City, California, USA, July 1-3, 2012.
52. 12th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'12), Stockholm, Sweden, 13-16 June 2012.
53. 15th IEEE International Symposium on Object and component-oriented Real-time distributed Computing (ISORC), April 11-13, 2012, Shenzhen, China.
54. 23rd IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2011), Dallas, USA, December 14 to 16, 2011.
55. Fourth IEEE International Workshop on Real-Time Service-Oriented Architecture and Applications (RTSOAA 2011), December 12th–14th 2011, University of California, Irvine, CA.
56. ACM/IFIP/USENIX International Middleware Conference, Lisbon, Portugal, December 12th to 16th, 2011.
57. 9th International Conference on Advances in Mobile Computing and Multimedia (MoMM2011), Hue City, Vietnam, 05-07 December 2011.
58. Control Systems, Automation and Robotics track of the 3rd International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2011), Hungary on October 5-7, 2011.
59. 15th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2011), August 29th - September 2nd, 2011, Helsinki, Finland.
60. 15th International Software Product Line Conference (SPLC 2011), Research/Experience Track, Munich, Germany, August, 22-26, 2011.
61. 15th International Software Product Line Conference (SPLC 2011), Industry Track, Munich, Germany, August, 22-26, 2011.
62. 2nd Workshop on Formal Methods in Software Product Line Engineering - Munich (Germany), August 2011.
63. 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE2011), Miami Beach, USA, July 7-9, 2011.
64. 2nd International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems, July, 5th 2011, Porto, Portugal.
65. Fourth IEEE International workshop UML and Formal Methods, co-located with FM 2011, June 20th, 2011, Lero, Limerick, Ireland.

66. The Software Engineering and Data Engineering (SEDE 2011) conference, Las Vegas, Nevada, June 20-22, 2011.
67. 3rd International Workshop on Model-Driven Architecture and Modeling-Driven Software Development (MDA&MDSD 2011) in conjunction with the 6th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE 2011, Beijing Jiaotong University, 8-11, June 2011.
68. 11th International IFIP Conference on Distributed Applications and Interoperable Systems (DAIS 2011), Reykjavik, Iceland, June 6-9 2011.
69. Second Product LinE Approaches in Software Engineering (PLEASE) workshop, collocated with 33rd International Conference on Software Engineering, Waikiki, Honolulu, Hawaii, May 21-28, 2011.
70. 16th Annual IEEE International Conference on the Engineering of Complex Computer Systems (ICECCS), April 27th-29th, 2011 Las Vegas, NV, USA.
71. Sixth IEEE International workshop UML and AADL, in conjunction with ICECCS 2011, April 27th, 2011, Las Vegas, USA.
72. First International Workshop on Cyber-Physical Networking Systems (CPNS'2011), in conjunction with INFOCOM 2011, April 15, 2011, Shanghai, China.
73. 2nd Workshop on Model Based Engineering for Embedded System Design (M-BED 2011), colocated with the Design, Automation, and Test in Europe (DATE) conference, 14-18, March, 2011, Grenoble, France.
74. Second International Conference on Emerging Applications of Information Technology (EAIT 2011), February, 2011 at Kolkata, India.
75. Fifth International Workshop on "Variability Modeling of Software-intensive Systems" (VaMoS '11), January 27-29 2011 in Namur, Belgium.
76. 9th Workshop on Adaptive and Reflective Middleware (ARM 2010) November 27, 2010, Bangalore India, collocated with Middleware 2010.
77. The 22nd IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2010), November 8-10, 2010, Marina Del Ray, California.
78. International Conference on Software Engineering, Management, and Application (ICSEMA 2010) Kathmandu, Nepal, October 29th and 30th, 2010.
79. The MobiCPS 2010 workshop, held in conjunction with the 7th International Conference on Ubiquitous Intelligence and Computing (UIC2010), October 26-29, 2010 Xian, China.
80. Fourteenth IEEE International Enterprise Computing Conference (EDOC 2010), 25-29 October 2010, Vitoria, ES, Brazil.
81. Advances in Business ICT (ABICT) 2010 Workshop Wisla, Poland, October 18-20, 2010.
82. 3rd Workshop on Model Based Architecting and Construction of Embedded Systems (ACES-MB), held in conjunction with MoDELS 2010, Oslo, Norway, October 3-8, 2010.
83. 4th Dynamic Software Product Line Workshop held in conjunction with the 14th International Software Product Line Conference 2010, Jeju Island, South Korea, September 13-17, 2010.
84. TOOLS Europe 2010, Malaga, Spain, June 28 to July 2, 2010.
85. 22nd International Conference on Software Engineering and Knowledge Engineering (SEKE'2010), to be held July 1-3, 2010, Redwood City, California.
86. 13th International Symposium on Component Based Software Engineering (CBSE 2010), June 23-25 2010 in Prague, Czech Republic.
87. Sixth European Conference on Modelling Foundations and Applications (ECMFA), University of Pierre & Marie Curie, Paris, France. June 15-18, 2010.
88. 10th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS), Amsterdam, The Netherlands, June 7-9, 2010.
89. The 11th OMG Real-time/Embedded CORBA workshop, Washington DC, May 24-26, 2010.

90. Industrial track at the 32nd International Conference on Software Engineering (ICSE 2010), Cape Town (South Africa), May 2-8, 2010.
91. Thirteeth International Conference on Business Information Systems (BIS 2010), Berlin, Germany, May 3-5 2010.
92. 1st International Workshop on Product LinE Approaches in Software Engineering, May 2nd, 2010, Cape Town, South Africa, held in conjunction with the 32nd International Conference on Software Engineering (ICSE 2010).
93. Workshop on Effective Multicasting for Future Critical Networked Systems (EMFINES 2010), at the Eighth European Dependable Computing Conference (EDCC), Valencia, Spain, April 28-30, 2010.
94. 1st Workshop on Model-Based Engineering for Embedded Systems Design, co-located with DATE 2010, March 12, 2010 in Dresden, Germany.
95. IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2010), Oxford 22-26, March 2010.
96. Special session on “Advanced Peer-to-Peer Protocols and Applications” at the Ninth IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2010) February 16-18, 2010 Innsbruck, Austria.
97. Fourth Variability Modelling of Software-intensive Systems (VaMoS ’10) workshop, Linz, Austria - January 27-29, 2010.
98. 8th Workshop on Adaptive and Reflective Middleware (ARM’09), in collocation with the 10th ACM/IFIP/USENIX Middleware Conference, in Urbana Champaign, Illinois, November 30th, 2009.
99. Workshop committee for OOPSLA 2009, Orlando Floria, October 25-29, 2009.
100. The ARTIST 2nd International Workshop on Model Based Architecting and Construction of Embedded Systems (ACESMB 2009), in conjunction with the 12th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2009), October 6th, 2009, Denver, Colorado.
101. The Thirteenth IEEE International EDOC Conference (EDOC 2009), 31 August - 4 September 2009, Auckland, New Zealand.
102. The 10th OMG Real-time/Embedded CORBA workshop, Washington DC, July 13–15, 2009.
103. The Software Engineering and Knowledge Engineering (SEKE’2009) conference, July 1-3, 2009, Boston, MA.
104. 12th International Symposium on Component-Based Software Engineering (CBSE 2009), East Stroudsburg University, Pennsylvania, USA, June 22-25, 2009.
105. The Second International Workshop on Cyber-Physical Systems (WCPS2009), held in conjunction with IEEE ICDCS 2009 in Montreal, Canada, June 22, 2009.
106. The Fifth European Conference on Model Driven Architecture Foundations and Applications (ECMDA), Gdansk, Poland, summer of 2009.
107. The 9th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2009) conference, Lisbon, Portugal, June 9-11, 2009.
108. The Fourth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE), 15th - 19th June 2009, Trinity College Dublin, Ireland.
109. The UML&AADL Workshop, held in conjunction with ICECCS 2009 The fourteenth IEEE International Conference on Engineering of Complex Computer Systems June 02, 2009, Potsdam, Germany.
110. The 15th Real-time and Embedded Applications Symposium (RTAS) 2009, Track B, Real-time and Embedded Applications, Benchmarks and Tools, San Francisco, CA, United States, April 13 - 16, 2009.
111. Member of the ISORC 2009 advisory and publicity committee for ISORC 2009, March 17-20, 2009, Toyko, Japan.

112. the 13th International Software Product Line Conference (SPLC), August 24-28, 2009, San Francisco, CA.
113. the European Conference on Model Driven Architecture - Foundations and Applications 2009, University of Twente, Netherlands, June 2009.
114. The third workshop on "Variability Modelling of Software-intensive systems" (VaMoS'09), January 28-30 2009 in Sevilla, Spain.
115. the 1st Workshop on Software Reuse Efforts, November 27-29, 2008 Brazil.
116. the 7th Workshop on Adaptive and Reflective Middleware (ARM'08) in collocation with the 9th ACM/IFIP/USENIX Middleware Conference, Leuven, Belgium, December 1st 2008.
117. the Middleware 2008 9th International Middleware Conference, December 1-6, 2008, Leuven, Belgium.
118. the 11th Component-Based Software Engineering conference, Karlsruhe, Germany, October 14-17, 2008.
119. the ARTIST International Workshop on Model Based Architecting and Construction of Embedded Systems (ACESMB 2008), in conjunction with the 11th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2008), Toulouse, September 29th, 2008.
120. the 6th Java Technology for Real-Time and Embedded Systems (JTRES) conference, Santa Clara, California, USA, 24-26 September, 2008.
121. the 12th IEEE International Enterprise Distributed Computing Conference (EDOC) (EDOC 2008), 15-19 September 2008, Munich, Germany.
122. the First Workshop on Analyses of Software Product Lines (ASPL'08), September 12, 2008 in Limerick, Ireland in conjunction with SPLC'08.
123. the 9th OMG Real-time/Embedded CORBA workshop, Washington DC, July 14–17, 2008
124. the 3rd International Conference on Software and Data Technologies, July 5-8, 2008, Porto, Portugal.
125. the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08), Redwood City, California, USA, July 1-3, 2008.
126. the TOOLS EUROPE 2008 conference, June 30 to July 4, 2008 at ETH Zurich.
127. National Conference on Research & Development in Hardware & Systems (CSI-RDHS 2008), Computer Society of India Kolkata Chapter & CSI Division I (Hardware & Systems), June 20-21, 2008, Kolkata, India.
128. the First International Workshop on Cyber-Physical Systems, Beijing, China, June 17 - 20, 2008.
129. the ECMDA 2008 (Fourth European Conference on Model Driven Architecture Foundations and Applications) in Berlin, June 09 - 12, 2008.
130. the Distributed Applications and Interoperable Systems (DAIS), Oslo, Norway, June 4, 2008.
131. the 2nd International Workshop on Ultra-Large-Scale Software-Intensive Systems (ULSSIS 2008), May 10-11, 2008 Leipzig, Germany.
132. the Automotive Systems Track at the 30th International Conference on Software Engineering (ICSE), Leipzig, Germany, 10-18 May 2008.
133. the Real-Time and embedded Applications / Benchmarks track at the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008), St. Louis, MO, April 22-24, 2008.
134. the 3rd UML and AADL Workshop held in conjunction with the 13th IEEE International Conference on Engineering of Complex Computer Systems, Belfast, Northern Ireland, 31 March - 4 April 2008.
135. the ACM Programming for Separation of Concern track at SAC 2008, Fortaleza, Brazil, March 16 - 20, 2008.
136. the 6th edition of the International Workshop on Adaptive and Reflective Middleware, held in conjunction with Middleware 2007 in Newport Beach, California.

137. the IEEE/ACM/USENIX Middleware conference, November 2007.
138. the IASTED International Conference on Parallel and Distributed Computing and Systems, PDCS 2007, Cambridge, MA, USA from Nov 19-21, 2007.
139. the 9th International Symposium on Distributed Objects, Middleware, and Applications (DOA), Iberian peninsula and islands, Oct 28 - Nov 2, 2007.
140. Member of the Doctoral Symposium committee at OOPSLA 2007, Portland, OR October 21-25, 2007.
141. the International Symposium on Ambient Intelligence and Computing, October 2007, Korea.
142. the IEEE conference on Enterprise Distributed Object Computing (EDOC), Annapolis, MD, October 15-19, 2007.
143. the 5th Java Technology for Real-Time and Embedded Systems (JTRES), Vienna, Austria, 26-28 September, 2007.
144. the Workshop on Trade-Off analysis of Software Quality Attributes (TOSQA), collocated with the sixth joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Dubrovnik, Croatia, September 3-7, 2007.
145. the 2nd International Conference on Software and Data Technologies, July 22-25th 2007, Barcelona, Spain.
146. the Fourth IEEE International Conference on Web Services, Salt Lake City, UT, July 9-13, 2007.
147. the 10th International Component-Based Software Engineering (CBSE) Symposium, Boston, MA, July 9-11 2007.
148. the 8th OMG Real-time/Embedded CORBA workshop, Washington DC, July 9–12, 2007.
149. the International Conference TOOLS EUROPE 2007, Zurich, Switzerland on June, 24-28 2007.
150. the track on "Real-Time and Embedded Applications and Benchmarks" for the 13th IEEE Real-Time and Embedded Technology and Applications Symposium, Bellevue, WA, April 3 - April 6, 2007.
151. the Workshop on the Foundations of Interactive Computation (FInCo 2007), Braga, Portugal, March 24 - April 1, 2007.
152. the 15th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Long Beach, California, 26-27 March, 2007.
153. the ACM Symposium on Applied Computing, Programming for Separation of Concerns track, Seoul, Korea, March 11 - 15, 2007.
154. the Workshop on Pervasive Computing Environments and Services (PCES 07), Naples, Italy, Feb 7-9, 2007.
155. the Minitrack on Components for Embedded and Real-time Systems at the 40th Hawaiian International Conference on System Sciences, January 3-6, 2007 at Waikoloa, Big Island, Hawaii.
156. the 13th Asia Pacific Software Engineering Conference (APSEC06), Bangalore, India, Dec 6-8, 2006.
157. the Real-time Middleware and Software Engineering track of the The 27th IEEE Real-Time Systems Symposium, December 5-8, 2006 Rio de Janeiro, Brazil.
158. the 2nd International Conference on Trends in Enterprise Application Architecture, November 29th to December 1st, 2006, Berlin, Germany.
159. the workshop on MOdel Driven Development for Middleware (MODDM), November 27, 2006, Melbourne, Australia.
160. the International Symposium on Distributed Objects and Applications (DOA), Montpellier, France, Oct 29 - Nov 3, 2006.
161. the "Library-Centric Software Design" (LCSD'06) workshop at the OOPSLA'06 conference in Portland, Oregon, October 22-26, 2006.
162. Judge for the Student Research Competition at OOPSLA 2006, Portland, OR, October 23-24, 2006.

163. the NSF Workshop On Cyber-Physical Systems, October 16 - 17, 2006, Austin, Texas.
164. the Models at Run-Time MaRT-06 workshop held at the MoDELS 2006 conference, Genova Italy, Oct. 2-6, 2006.
165. the MoDELS 2006 conference, Genova Italy, Oct. 2-6, 2006.
166. the 7th OMG Real-time/Embedded CORBA workshop, Washington DC, July 11–14, 2006.
167. the European Conference on Object-Oriented Programming, Nantes, France, July 3-7, 2006.
168. the 9th International Symposium on Component-Based Software Engineering (CBSE 2006), Malardalen University, Sweden, June 29th-1st July 2006.
169. the 28th International Conference on Software Engineering (ICSE 28), May 24-26, 2006, Shanghai, China.
170. the 14th International Workshop on Parallel and Distributed Real-Time Systems, April 25-26, 2006, Island of Rhodes, Greece.
171. the 9th IEEE International Symposium on Object-oriented Real-time Distributed Computing, April 24-26, 2006, Gyeongju, Korea.
172. the Automotive Software Workshop San Diego (ASWSD 2006), University of California, San Diego, March 15-17, 2006.
173. the C++ Connections: 20 Years of C++ conference, Nov 7-11, 2005, Mandalay Bay, Las Vegas, NV.
174. the Conference on Distributed Objects and Applications (DOA 2005), Oct 31 - Nov 4, 2005, Agia Napa, Cyprus.
175. the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOSPLA), October 16-20, 2005, San Diego, CA, USA.
176. the 6th International Conference on Middleware (Middleware'2005), October, 2005, Grenoble, France.
177. the 2005 Monterey Workshop on Networked Systems, Laguna Beach, California, September 22-24, 2005.
178. The 12th Pattern Language of Programs (PLoP 2005), September 7-10, 2005, Allerton Park, Monticello, Illinois, USA.
179. the 14th IEEE International Symposium on High-Performance Distributed Computing (HPDC-14), Research Triangle Park, North Carolina, July 27, 2005.
180. the 5th International Workshop on Software and Performance (WOSP 2005), Palma de Mallorca, Spain, July 11-15, 2005.
181. the 6th OMG Real-time/Embedded CORBA workshop, Washington DC, July 11–14, 2005.
182. the 5th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS 2005), June 15-17, 2005, Athens, Greece.
183. the International Conference on Autonomic Computing (ICAC 2005), Seattle, WA, June 2005.
184. the International Symposium on Component-Based Software Engineering (CBSE), co-located with the International Conference on Software Engineering (ICSE), May 14-15, 2005, St. Louis, MO.
185. the Foundations of Interactive Computation (FINCO'05) Workshop, Saturday, 9 April 2005, in Edinburgh, Scotland.
186. the Embedded Applications track of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2005, San Francisco, California, March 2005.
187. the “Programming for Separation of Concerns” track at Symposium on Applied Computing (SAC 2005), Santa Fe, New Mexico, March 2005.
188. the 12th International Symposium on the Foundations of Software Engineering, November 6th, 2004, Newport Beach, California.
189. the Conference on Distributed Objects and Applications (DOA 2004), October 25-29, 2004 in Cyprus, Greece.

190. the 2nd International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES), October 25-29, 2004, Larnaca, Cyprus.
191. the 3rd Workshop on Reflective and Adaptive Middleware (RM2004), October 19, 2004, Toronto, Ontario, Canada.
192. the Middleware 2004 5th IFIP/ACM/USENIX International Conference on Distributed Systems Platforms, October 18-22, 2004, Toronto, Canada.
193. the 4th TAO+CIAO Workshop, Arlington, VA, July 16, 2004.
194. the DARPA Workshop on Java in Real-Time and Embedded Defense Applications, Arlington, VA, July 13, 2004.
195. the OMG Real-time/Embedded CORBA workshop, Crystal City, VA, July 12–15, 2004.
196. the ECOOP 2004 conference, June 14-18, 2004, Oslo, Norway.
197. the Middleware track of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS), May 23-26, 2004, Tokyo, Japan.
198. the 2nd International Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), Edinburgh, Scotland, UK, May 24, 2004.
199. Aspect-Oriented Software Development conference, Lancaster, England, March 22-26, 2004.
200. the SPIE/ACM Conference on Multimedia Computing and Networking, January 21-22, 2004 Santa Clara, California.
201. the Real-time Systems Symposium (RTSS), Cancun, Mexico, December 3-5, 2003.
202. the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), Paris - France November 17-21, 2003.
203. the International Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES), November 3-7, 2003, Catania, Sicily, Italy.
204. the Domain Driven Development track at the OOPSLA 2003 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, October 26-30, 2003, Anaheim, California, USA.
205. the OOPSLA 2003 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, October 26-30, 2003, Anaheim, California, USA.
206. External reviewer for the 2nd Generative Programming and Component Engineering (GPCE '03) conference, Erfurt, Germany, September 22-25, 2003.
207. the OMG Real-time/Embedded CORBA workshop, Crystal City, VA, July 14–17, 2003.
208. the The 2nd Workshop on Reflective and Adaptive Middleware, Rio de Janeiro, Brazil, June 17, 2003.
209. the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI), San Diego, California, June 9 - 11, 2003.
210. the 1st International Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), Portland, Oregon, May 9, 2003.
211. External reviewer for the 17th International Parallel and Distributed Processing Symposium, April 22–26, 2003, Nice, France.
212. the ACM International Conference on Aspect-Oriented Software Development, March 17 - 21, 2003, Boston, MA.
213. the SPIE/ACM Conference on Multimedia Computing and Networking, Santa Clara, California, January 29–31, 2003.
214. the International Workshop on Product Line Engineering The Early Steps: Planning, Modeling, and Managing (PLEES '02), Seattle, WA, November 5, 2002.
215. the 8th IEEE Real-Time and Embedded Technology and Application Symposium (RTAS), San Jose, CA, September 24-27, 2002.
216. the 9th Conference on Pattern Language of Programs, Allerton Park, IL, September 8-12, 2002.

217. the Workshop on Dependable Middleware-Based Systems, held as a part of DSN 2002, Washington, D.C., June 23-36, 2002.
218. the 2nd TAO Workshop, Arlington, VA, July 19, 2002.
219. the OMG Real-time/Embedded CORBA workshop, Crystal City, VA, July 15–18, 2002.
220. the 16th European Conference on Object-Oriented Programming, University of Malaga, Spain June 10-14, 2002.
221. the Tenth International Workshop on Quality of Service (IWQoS), May 15-17, 2002, Miami Beach, Florida.
222. the International Symposium on Object-Oriented Real-time Distributed Computing (ISORC), Washington DC, April 29 – May 1, 2002.
223. the Seventh IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS 2002), January 7-9, 2002, San Diego, CA.
224. the International Workshop on Multimedia Middleware October 5th, 2001, Ottawa, Canada.
225. the OMG Workshop on Real-time and Embedded CORBA, in Reston, VA, June 4-6, 2001.
226. the USENIX 2001 conference, Boston, MA, June 25-30, 2001.
227. the International Symposium on Object-oriented Real-time Distributed Computing (ISORC), May 2-4, Magdenburg, Germany, 2001.
228. the 6th USENIX Conference on Object-Oriented Technologies and Systems, January 27 - February 3, 2001, San Antonio, TX.
229. External reviewer for OOPSLA 2000, Minneapolis, MN, October 2000.
230. the 3rd IFIP International Conference on Trends towards a Universal Service Market (USM'2000), September 12-14, 2000.
231. the International Symposium on Distributed Objects and Applications (DOA '00), OMG, Antwerp, Belgium, September 2000.
232. the ACM SIGCOMM 2000, Stockholm, Sweden, August 30 to September 1st, 2000.
233. the Pattern Languages of Programming (PLoP) conference, Monticello, Illinois, August, 2000.
234. the 9th IEEE International Conference on High-Performance Distributed Computing, August, 2000.
235. the "International Workshop on Software Engineering for Parallel and Distributed Systems" (PDSE 2000), at the 22nd International Conference on Software Engineering (ICSE-2000), in Limerick, Ireland in June, 2000.
236. the 6th IEEE Real-Time Technology and Application Symposium (RTAS), May 17-19, 2000, Washington DC, USA.
237. the 1999 ACM OOPSLA conference, Denver, Colorado, November 1-5, 1999.
238. the IFIP Sixth International Workshop on Protocols For High-Speed Networks (PfHSN '99), Wednesday August 25 – Friday August 27, 1999 Salem, MA.
239. the 1999 IEEE Real-Time Technology and Applications Symposium (RTAS99), Vancouver, British Columbia, Canada, June 2-4, 1999.
240. the 5th USENIX Conference on Object-Oriented Technologies and Systems, May 3-7, 1999, San Diego, CA.
241. Technical workshop committee for the International Software Architecture workshop, ACM SIGSOFT's FSE9 conference in Orlando FL, November 1-5, 1998.
242. the workshop on Software and Performance (WOSP98), Santa Fe, New Mexico, Oct 12-16 1998.
243. the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing: Middleware '98. September 15-18 1998, The Lake District, England.
244. the TOOLS USA'98 conference. Santa Barbara, California, August 3 - 7, 1998.
245. the IEEE High Performance Distributed Computing conference, Chicago, IL, July 28-31, 1998.
246. $12^{th}$ European Conference on Object-Oriented Programming, Brussels, Belgium, July 20 - 24, 1998.
247. the 3rd EuroPLoP conference, Kloster Irsee, Germany, July 9-11, 1998.

248. the IEEE International Conference on Configurable Distributed Systems (ICCDS '98), Annapolis, MD, May 4-6, 1998.
249. the IEEE IWQoS '98 in Napa Valley, CA, May 18-20, 1998.
250. the 4th USENIX Conference on Object-Oriented Technologies and Systems, April 26-29, 1998, Santa Fe, New Mexico.
251. the $3^{rd}$ International Workshop on Software Engineering for Parallel and Distributed Systems, at the 20th International Conference on Software Engineering (ICSE-20), in April 20-21, Kyoto, Japan.
252. the IEEE Conference on Open Architectures and Network Programming, April 3-4, 1998, San Francisco, CA.
253. the Workshop on Middleware for Real-Time Systems and Services, held in conjunction with IEEE Real-time Systems Symposium, December 2nd, San Francisco, California.
254. the Open Signaling for ATM, Internet and Mobile Networks. October 6th and 7th, 1997, Columbia University, New York, NY.
255. the $24^{th}$ International Conference on Technology of Object-Oriented Languages and Systems (TOOLS Asia '97). Beijing, China, September 22 - 25, 1997.
256. the $4^{th}$ Pattern Languages of Programming conference, Allerton Park, Illinois, September 3-5, 1997.
257. the $3^{rd}$ USENIX Conference on Object-Oriented Technologies and Systems, Portland, June 16-19th 1997.
258. Session chair of the Patterns technical paper session at ECOOP '97, June 13th, 1997.
259. the 1997 European Conference on Object-Oriented Programming (ECOOP), June 9-13, 1997, Jyvskyl, Finland.
260. Chair of the technical session on "Distributed Object Computing" for the IFIP/IEEE Fifth International Workshop on Quality of Service (IWQoS '97).
261. the $2^{nd}$ International Workshop on Software Engineering for Parallel and Distributed Systems, at the $19^{th}$ International Conference on Software Engineering (ICSE-19) Sheraton Boston Hotel and Towers, Boston, Massachusetts, USA, May 19 and 20, 1997.
262. the $3^{rd}$ USENIX Conference on Object-Oriented Technologies and Systems, Portland, 1997.
263. the $5^{th}$ IEEE International Workshop on Object-Orientation in Operating Systems, IEEE TCOS and USENIX, Seattle, Washington, October 27-28, 1996.
264. the 1997 ACM SIGCOMM conference, Cannes, French Riviera, France, September 1997.
265. the 1997 IEEE INFOCOM conference, Kobe, Japan, April 1997.
266. the 1996 IEEE INFOCOM conference, San Francisco, CA, USA, March 24-28, 1996.
267. the 1995 IEEE INFOCOM conference, Boston, Massachusetts, USA, April, 1995.
268. the $3^{rd}$ IEEE workshop on Architecture and Implementation of High Speed Communication Subsystems (HPCS '95), held in Mystic, Connecticut, August 1995.
269. the $8^{th}$ IFIP International Working Conference on Upper Layer Protocols, Architectures, and Applications, held in Barcelona, Spain, June 1 to 3, 1994.

**Workshops and Panels Organized**

1. Co-organized the 1st International Workshop on Data Dissemination for Large scale Complex Critical Infrastructures (DD4LCCI 2010), at the Eighth European Dependable Computing Conference, Valencia, Spain, April 28-30, 2010.
2. Co-organized the OOPSLA Jeopardy panel at OOPSLA 2009, Orlando Floria, October 25-29, 2009.
3. Co-organized a workshop entitled First International Workshop on Software Technologies for Ultra-Large-Scale (ULS) Systems at 29th Int. Conference on Software Engineering, May 20-29th, Minneapolis, MN, 2007.
4. Co-organized a session on architectures, platforms, and standards for QoS-enabled dissemination at the Systems and Information Interoperability Meeting, Oct 25-27, 2006 at the Minnowbrook Conference Center, Blue Mountain Lake, NY.

5. Co-organized a workshop entitled "Breathturn: Ultra Large Scale Systems" at OOPSLA 2006, October 26, 2006, Portland, OR.
6. Co-chair of the NSF workshop on open-source Middleware for Distributed Real-time and Embedded Systems, 7th OMG Real-time/Embedded CORBA workshop, Arlington, VA, July 10–13, 2006.
7. Organized and led a session on architectures, platforms, and standards for real-time tactical information management at the Systems and Information Interoperability Meeting, Oct 18-21, 2005 at the Minnowbrook Conference Center, Blue Mountain Lake, NY.
8. Co-organizer of the technical workshops program at OOPSLA 2005, San Diego, October 16th-20, 2005.
9. Co-organizer for the MODELS 2005 workshop on "MDD for Software Product-lines: Fact or Fiction?," October 2, 2005, Jamaica.
10. Co-organizer of the OOPSLA '02 workshop on "Patterns in Distributed Real-Time and Embedded Systems", Seattle, WA, November, 2002.
11. Co-organizer of the OOPSLA '01 workshop on "Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems" Tampa Bay, FL, October 14, 2001.
12. Organizer and chair of a panel on real-time extensions to OO middleware, OPENSIG Fall '97 workshop on Open Signaling for ATM, Internet and Mobile Networks Columbia University, October 6-7 1997, New York, NY.
13. Co-organizer of a workshop for the 1997 European Conference on Object-Oriented Programming entitled CORBA: Implementation, Use, and Evaluation, Jyvaskyla, Finland, June 10th, 1997.
14. Organizer and chair of a panel on "QoS and Distributed Systems Platforms" for the IFIP Fifth International Workshop on Quality of Service (IWQoS '97), May 22-24th, 1997, Columbia University, New York.
15. Co-organizer of the OOPSLA '95 workshop on "Patterns for Concurrent, Parallel, and Distributed OO Systems."
16. Co-facilitator of the ECOOP '95 workshop workshop on Pattern Languages of Object-Oriented Programs, Aarhus, Denmark, August 1995.

**Reviewer for Professional Submittals**

Reviewed papers for the following journals, conferences, books, and grant review processes:

1. Reviewer for COVID-19 proposals to the C3.ai Digital Transformation Institute.
2. *The 21st IEEE International Symposium on Real-time Computing (ISORC)*, Nanyang Technological University, Singapore, 29th - 31st May 2018.
3. *Future Generation Computer Systems*, Elsevier, edited by Aniruddha Gokhale et al., 2016.
4. *IEEE Software*, Special Issue on Next Generation Mobile Computing, edited by James Edmondson et al., 2013.
5. *Software Testing in the Cloud*, edited by Scott Tilley, 2012.
6. Elsevier Information & Software Technology special issue on Software Reuse and Product Lines, 2012.
7. The 2010 Military Communications Conference, Cyber Security and Network Management, San Jose, CA, October 31-November 3, 2010.
8. *Model-Driven Domain Analysis and Software Development: Architectures and Functions*, edited by Janis Osis and Erika Asnina, 2010.
9. Reviewer for the book "Patterns for Parallel Software Design," by Jorge L. Ortega Arjona, Wiley, 2010.
10. Special Issue on Industrial Applications of Aspect Technology for the journal Transactions on Aspect-Oriented Software Development (TAOSD), 2009.
11. *Software Engineering for Self-Adaptive Systems*, edited by Betty H. C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, Springer, 2009.
12. Special issue on Service Oriented Computing for the ACM Transactions on the Web journal, 2008.

13. Special Issue in Software Reuse: Methods, Processes, Tools and Experiences for the Journal of the Brazilian Computer Society (JBCS), 2007
14. Designing Software-Intensive Systems: Methods and Principles book, 2008
15. Special issue on Patterns for the IEEE Software, 2007
16. IEEE Internet Computing Magazine, 2006.
17. IEEE Transactions on Parallel and Distributed Systems, 2004
18. International Journal of Software Process: Improvement and Practice Special issue - Software Variability: Process and Management
19. IEEE Internet Computing Magazine
20. 2004 NSF NSG panel
21. IEEE Transactions on Parallel and Distributed Computing special issue on Middleware, 2003
22. 2003 NSF ITR panel
23. 2002 NSF CAREER panel
24. IEEE Internet Computing Magazine, 2002
25. NIST Competence Proposals, May 2002
26. DARPA MoBIES program, May 2002
27. DARPA NEST program, May 2002
28. DARPA DASADA program, April 2002
29. Elsevier Journal of Systems and Software Special Issue on Software Architecture: Engineering Quality Attributes, 2002
30. IEEE Communications Magazine, Evolving Communications Software: Techniques and Technologies, 2001
31. DARPA Network Embedded Software Technology (NEST) program, 2001
32. DARPA Software Enabled Control (SEC) program, 2000
33. IEEE Concurrency magazine, Object-Oriented Systems Track, 1999
34. IEEE Journal on Selected Areas in Communications special issue on "Service Enabling Platforms for Networked Multimedia Systems," 1999
35. IEEE Journal of Communications and Networks, 1999
36. Reviewer for the $4^{th}$ Pattern Languages of Programming Design book published by Addison Wesley
37. The International Journal of Time-Critical Computing Systems, special issue on Real-time Middleware, edited by Wei Zhao
38. Next Generation Internet (NGR) networking research review panel, October 1998
39. IEE Transactions on Software Engineering, special issue on Configurable Distributed Systems
40. Theme issue on Symbolic Modeling in Practice for the Communications of the ACM
41. "Multimedia DBMS and the WWW" Minitrack at the 32nd Hawaii International Conference on System Sciences, 1999
42. "Dependable Distributed Systems" Minitrack at the 32nd Hawaii International Conference on System Sciences, 1999
43. IEEE Computer special issue on "Design Challenges for High-Performance Network Interfaces," 1998
44. 1998 NSF Experimental Software Systems review panel.
45. ACM SIGMetrics Conference, 1998
46. ACM Transactions on Software Engineering Methods
47. Special Issue on Patterns and Pattern Languages for the journal of Theory and Practice of Object Systems, (Stephen P. Berczuk, Editor), John Wiley and Sons, 1995
48. Special Issue of Computer Communications on Building Quality of Service into Distributed Systems

49. IEEE Communications Magazine
50. IEEE/ACM Journal of Transactions on Networking
51. Communications of the ACM
52. IEE/BCS Distributed Systems Engineering Journal
53. Software Practice and Experience, John Wiley and Sons
54. 1998, 1997, and 1996 NSF networking program
55. 1996 NSF software engineering and programming languages CAREER panel
56. 1994 California MICRO (Microelectronics Innovation Computer Research Opportunity) engineering computer network grant review process
57. IEEE Conference on Parallel and Distributed Computing Systems, 1994
58. IEEE International Conference on Computer Communications and Networks, 1994
59. IEEE INFOCOM conference, 1994
60. 1993 NASA Applied Information Systems Research grant review process
61. 1992 California MICRO (Microelectronics Innovation Computer Research Opportunity) engineering computer network grant review process
62. $7^{th}$ *IFIP International Conference on Upper Layer Protocols, Architectures, and Applications*, 1992
63. The 1992 Special Issue on Measurement for IEEE Journal Transactions on Software Engineering

**Memberships**: IEEE, ACM, and USENIX

# Patents

1. US patent 7,523,471 – "Interpretive network daemon implemented by generic main object," in conjunction with Karlheinz Dorn, Dieter Quehl, Detlef Becker, and Christian Scharf of SIEMENS Medical Engineering, Erlangen, Germany, 2009.

# Theses Supervised

- *Doctoral and Masters Committees Chaired*

  1. Chaired the MS thesis committee for Evan Segaul, March 2021.
  2. Co-chair of the doctoral dissertation defense for Peng Zhang, August 2018.
  3. Co-chair of the doctoral dissertation defense for James Edmondson, March 2012.
  4. Co-chair of the doctoral topic defense for James Edmondson, December 2011.
  5. Co-chair of the doctoral dissertation defense for Will Otte, November 2011.
  6. Chair of the doctoral dissertation defense for Brian Dougherty, March 2011.
  7. Chair of the doctoral topic defense for Brian Dougherty, June 2010.
  8. Chair of the masters defense for Pooja Varshneya, May 2010.
  9. Chair of the doctoral topic defense for Nilabja Roy, March 2010.
  10. Chair of doctoral topic defense for Joe Hoffert, November 2009.
  11. Chair of the doctoral dissertation defense for Jai Balasubramanian, September 2009.
  12. Chair of masters defense for Friedhelm Wolf, March 2009.
  13. Chair of the doctoral dissertation defense for Nishanth Shankaran, October 2008.
  14. Chair of the doctoral dissertation defense for Jules White, October 2008.
  15. Chair of doctoral dissertation defense for Gan Deng, December 2007.
  16. Chair of doctoral dissertation defense for Krishnakumar Balasubramanian, September 2007.
  17. Chair of the doctoral topic defense for Nishanth Shankaran, April 2007.
  18. Chair of doctoral topic defense for Krishnakumar Balasubramanian, March 2006.
  19. Chair of doctoral topic defense for Gan Deng, March 2006.
  20. Chair of final doctoral dissertation defense for Arvind Krishna, December 2005.

21. Chair of MS thesis committee for Emre Turkay, summer 2005.
22. Chair of doctoral topic defense for Arvind Krishna, summer 2005.
23. Chair of MS thesis committee for Ossama Othman, December, 2002.
24. Chair of doctoral dissertation committee for Carlos O'Ryan, May, 2002.
25. Chair of dissertation topic defense committee for Carlos O'Ryan, September, 2001.
26. Chair of masters committee for Nagarajan Surendran, August, 1999.
27. Chair of masters committee for Alexander Babu Arulanthu, July, 1999.
28. Chair of oral exam committee for Chris Gill, June, 1999.
29. Chair of doctoral exam committee for Andy Gokhale, May, 1998.
30. Chair of masters exam committee for Sumedh Mungee, May, 1998.
31. Chair of masters exam committee for Sergio Flores, May, 1998.
32. Chair of masters committee for Prashant Jain, June 1997.
33. Chair of doctoral topic defense for James Hu, February 1997.
34. Chair of masters committee for Tim Harrison, February 1997.
35. Chair of doctoral topic defense committee for Andy Gokhale, October, 1996.

- *Doctoral and Masters Committees Member*

1. Served on the doctoral topic defense for Zhongwei Teng, April 2021.
2. Served on the MS thesis committee for Gabriela Gresenz, March 2021.
3. Served on the MS thesis committee for Xiaoxing Qiu, March 2021.
4. Served on the doctoral dissertation defense for Anirban Bhattacharjee, January 2020.
5. Served on the doctoral topic defense for Anirban Bhattacharjee, April 2019.
6. Served on the doctoral dissertation defense for Shunxing Bao, September 2018.
7. Served on the doctoral dissertation defense for Shashank Shekhar, May 2018.
8. Served on the doctoral dissertation defense for Fangzhou Sun, March 2018.
9. Served on the doctoral topic defense for Shunxing Bao, March 2018.
10. Served on the doctoral topic defense for Peng Zhang, January 2018.
11. Served on the doctoral dissertation defense for Marcelino Rodriguez-Cancio, December 2017.
12. Served on the doctoral dissertation defense for Yao Pan, November 2017.
13. Served on the doctoral topic defense for Fangzhou Sun, September 2017.
14. Served on the doctoral topic defense for Shashank Shekhar, May 2017.
15. Served on the doctoral topic defense for Yao Pan, February 2017.
16. Served on the doctoral dissertation defense for Faruk Caglar, July 2015
17. Served on the doctoral dissertation defense for Wei Yan, May 2015.
18. Served on the doctoral dissertation defense for Kyoungho An, March 2015.
19. Served on the MS thesis committee for Songtao Hei, March 2015.
20. Served on the MS thesis committee for Meng Wang, March 2015.
21. Served on the doctoral dissertation defense for Sean Hayes, January 2015.
22. Served on the doctoral dissertation defense for Hamilton Turner, November 2014.
23. Served on the doctoral topic defense for Faruk Caglar, November 2014.
24. Served on the doctoral topic defense for Hamilton Turner, February 2014.
25. Served on the doctoral dissertation defense for Fan Qui, February 2014.
26. Served on the doctoral dissertation defense for Xiaowei Li, May 2013.
27. Served on the doctoral topic defense for Fan Qiu, April 2013.
28. Served on the doctoral dissertation defense for Janos Mathe, August 2012.
29. Served on the doctoral dissertation defense for Tripti Saxena, July 2012.
30. Served on the doctoral dissertation defense for Akshay Dabholkar, April 2012.
31. Served on the doctoral topic defense for Xiawei Li, March 2012.
32. Served on the doctoral topic defense for Janos Mathe, August 2011.
33. Served on the doctoral dissertation defense for Liang Dai, April 2011.

34. Served on the doctoral dissertation defense for Daniel Balasubramanian, March 2011.
35. Served on the doctoral topic defense for Will Otte, February 2011.
36. Served on the doctoral topic defense for Akshay Dabholkar, February 2011.
37. Served on the doctoral dissertation defense for Joe Hoffert, February 2011.
38. Served on the doctoral topic defense for Tripti Saxena, January 2011.
39. Served on the doctoral dissertatin defense for Nilabja Roy, November 2010
40. Served on the doctoral topic defense for Daniel Balasubramanian, October 2010.
41. Served on the doctoral dissertation defense for Sumant Tambe, September 2010.
42. Served on the doctoral topic defense for Sumant Tambe, April 2010.
43. Served on the doctoral dissertation defense for John Kinnebrew, March 2010.
44. Served on the doctoral dissertation defense for Shanshan Jiang, November 2009.
45. Served on the doctoral dissertation defense for James Hill, March 2009.
46. Served on the doctoral topic defense for James Hill, October 2008.
47. Served on the doctoral topic defense for Jai Balasubramanian, August 2008.
48. Served on the doctoral topic defense for Liang Dai, December 2008.
49. Served on the doctoral topic defense for Shanshan Jiang, November 2008.
50. Served on the doctoral topic defense for Jules White, April 2008.
51. Served on the doctoral topic defense for Amogh Kavimandan, February 2008.
52. Served on the doctoral dissertation defense for Amogh Kavimandan, November 2008.
53. Served on the doctoral topic defense for Amogh Kavimandan, February 2008.
54. Served on the doctoral dissertation defense for Michael Stal, University of Groningen, March 2007.
55. Served on the doctoral topic defense for Karlkim Suwanmongkol, fall 2004.
56. Served on the doctoral dissertation topic defense committee for Aditya Agrawal, July, 2004.
57. Served on the doctoral dissertation defense for Angelo Corsaro, July 2004.
58. Served on the doctoral dissertation defense for Nanbor Wang, April 2004.
59. Served on the doctoral topic defense for Angelo Corsaro, October 2003.
60. Served on the doctoral dissertation defense committee for Jonathan Sprinkle, July, 2003.
61. Served on the doctoral dissertation topic defense committee for Aditya Agrawal, June, 2003.
62. Served on masters committee for Kirk Kelsey, March 2003.
63. Served on the dissertation topic defense committee for Jonathan Sprinkle, February, 2003.
64. Served as external examiner for Bob Jolliffe's masters thesis Department of Computer Science, University of South Africa, March, 2003.
65. Served on the doctoral dissertation committee for Irfan Pyarali, December, 2001.
66. Served on the doctoral dissertation committee for Chris Gill, December, 2001.
67. Served as external examiner for Daniel Heggander's Ph.D. dissertation in the Department of Software Engineering and Computer Science at Blekinge Institute of Technology, Sweden, September, 2001.
68. Served as external examiner for Mohammad Radaideh's masters thesis in the Electrical Engineering department at McMaster's University, Canada, Winter 2000.
69. Served as external examiner for David Holmes' Ph.D. dissertation in the information and computer sciences department at Macquarie University, Sydney, Fall 1999.
70. Served on final doctoral dissertation committee for Priya Narasimhan, August, 1999.
71. Served on the doctoral final dissertation defense for Christo Papadopoulos, August, 1999.
72. Served on dissertation topic defense for Michael Plezbert, February, 1999.
73. Served on masters committee for Craig Nauman, February, 1999.
74. Served on the doctoral exam committee for Chuck Cranor, July, 1998.
75. Served on masters exam committee for Mihai Tutunaru, April, 1998.
76. Served on the doctoral exam committee for Michael Plezbert, June, 1997.
77. Served on masters committee for Todd Rogers, June 1997.
78. Served on masters committee for Robert Engel, January 1997.

79. Served on committee for final doctoral dissertation defense of R. Gopalakrishnan, November, 1996.
80. Served on committee for final doctoral dissertation defense of Lorrie Cranor, September, 1996.
81. Served on the doctoral dissertation topic proposal committee for Christos Papadopoulos July, 1995.
82. Served on the doctoral dissertation topic proposal committee for Charles Cranor December, 1994.
83. Served on oral exam committee for Andy Gokhale December, 1994.
84. Served on the doctoral dissertation proposal committee for Lorrie Cranor, December, 1994.
85. Served on the doctoral final dissertation defense committee for Donald Wilcox, November, 1994.
86. Served on masters committee for Madhavapeddi Shreedhar, September, 1994
87. Served on the doctoral dissertation topic proposal committee for R. Gopalakrishnan, September, 1994.

- *Doctoral Student Advisees and Co-Advisees*

  1. Mike Walker (USA)

- *Graduated PhD Students*

  1. Jaiganesh Balasubramanian, Ph.D., 2009, currently works for Citigroup, New York, NY.
  2. Krishnakumar Balasubramanian, Ph.D., 2007, Mathworks, Boston, MA.
  3. Angelo Corsaro, Ph.D. 2004, PrismTechnologies, Parise France.
  4. Gan Deng, Ph.D., 2007, Citigroup, Charleston, SC.
  5. Brian Dougherty, Ph.D. 2011, Optio Labs, Nashville, TN.
  6. James Edmondson, Ph.D., 2012, Member of the Technical Staff, Software Engineering Institute, Pittsburgh, PA.
  7. Chris Gill, Ph.D. 2001, Professor, Washington University, St. Louis, MO.
  8. Andy Gokhale, Ph.D. 1998, Associate Professor, Vanderbilt University, Nashville, TN.
  9. James Hill, Ph.D., 2009, Assistant Professor, Indiana University, Purdue University, Indianapolis.
  10. Joe Hoffert, Ph.D. 2011, Assistant Professor, University of Edmonton, Canada.
  11. John Kinnebrew, Ph.D., 2010, ISIS, Nashville, TN.
  12. Arvind Krishna, Ph.D. 2005, Qualcomm, San Diego, CA.
  13. Irfan Pyarali, Ph.D. 2001, CitiGroup, New Jersey.
  14. Nilabja Roy, Ph.D. 2011, Research Scientist, Institute for Software Integrated Systems, Nashville, TN.
  15. Carlos O'Ryan, Ph.D., 2002, CitiGroup, Charleston, SC.
  16. Nishanth Shankaran, Ph.D., 2008, Amazon, Seattle, WA.
  17. Nanbor Wang, Ph.D. 2004, Research Scientist, Tech-X, Boulder, Colorado.
  18. Jules White, Ph.D. 2008, Assistant Professor, Virginia Tech, Blackburg, VA.

- *Graduated Masters and Ugrad Students*

  1. Alexander Babu Arulanthu, MS 1999, Sylantro, Campbell, CA.
  2. Everett Anderson, BS 1998, Sun, Mountain View, CA.
  3. Shawn Atkins, BS 1998, Lucent, Columbus, OH.
  4. Matt Braun, BS 1998.
  5. Darrell Brunsch, BS 1999, Microsoft, Redmond, WA.
  6. George Edwards, BS 2004, Ph.D. student at University of Southern California.
  7. Sergio Flores-Gaitan, MS 1998, Microsoft, Redmond, WA.
  8. Priyanka Gontla, MS 2000, UBS, Irvine, CA.
  9. Pradeep Gore, MS 2000, OOMWorks, New Jersey.
  10. Tim Harrison, MS 1997, Mayasoft, Palo Alto, CA.

11. Prashant Jain, MS 1997, IBM Research, India.
12. Vishal Kachroo, MS 1999, Stentorsoft, CA.
13. Michael Kircher, BS 1998, Siemens CT, Munich, Germany.
14. Yamuna Krishnamurthy, MS 2000, OOMWorks, New Jersey.
15. Tao Lu, MS 2003, Trading Technologies, Chicago, IL.
16. Sumedh Mungee, MS 1998, Fujitsu, Santa Clara, CA.
17. Bala Natarajan, MS 2000, Veritas, India.
18. Kirthika Parameswaran, MS 2000, Telcordia, Piscataway, NJ.
19. Stoyan Paunov, MS 2006, working at Bloomberg, NYC.
20. Ossama Othman, MS 2002, independent consultant, Portland, OR.
21. Marina Spivak, MS 2000, AT Desk, Charleston, SC.
22. Nagarajan Surendran, MS 1999, Sylantro, Campbell, CA.
23. Emre Turkay, MS 2005, Turkey.
24. Pooja Varshneya, May 2010, Zircon Computing, Wayne, NJ.
25. Seth Widoff, BS 1998, independent consultant, San Francisco, CA.
26. Ming Xiong, MS 2007, currently working at AT Desk, Charleston, SC.

- *Former Staff*

1. Chris Cleeland, OCI, St. Louis, MO.
2. Ray Klefstad, Research Assistant Professor, University of California, Irvine.
3. Boris Kolpackov, Independent Consultant, South Africa.
4. Fred Kuhns, Research Associate, Washington University, St. Louis, MO.
5. David Levine, Director of Engineering, CombineNet, Inc, Pittsburgh, PA.
6. Will Otte, Institute for Software Integrated Systems, Nashville, TN
7. Jeff Parsons, Optio Labs, Nashville, TN
8. Jules White, Ph.D. 2008, Vanderbilt University, Nashville, TN

## Research Support

Total research funding since June 1995: $41,899,342

- Sole PI: $12,030,403
- Co-PI: $29,868,939

### Grants and Contracts Received

1. "Automated Clothing Simulation and Human Avatar Generation Engine" NSF, 9/15/2019 to 2/29/2020, $50,000.
2. "Digital Thread Modelling Environment (DTME)," AFRL (subcontract through Securboration), 8/20/2019 to 8/20/2021, $250,000, with Jules White.
3. "Creating an Evidence-based Professional Development Support Tool for Pre-K Coaches and Teachers," Department of Education (IES), $1,399,992, 7/1/18 to 6/30/22, Co-PI with Caroline Christopher.
4. "Blockchain as Middleware Services for Transactive Energy Applications," Siemens, 4/1/2017 to 9/30/2018, $274,397, co-PI Abhishek Dubey.
5. "Children Eating Well (CHEW) Smartphone Application for WIC Families," USDA 4/15/2017 to 4/14/2022, , co-PI with Pam Hull.
6. "Industrial Internet Architecture," Varian Medical Systems, Inc., 10/1/14 to 12/31/18, $288,808, co-PI Jules White.
7. Securboration, "Virtualize Combat System Environment (ViCE)," $15,000, 1/1/18-3/26/18, Co-PI with Jules White.

8. "Container Hopping at Random Intervals or Targeted-Attacked (CHARIOT)," OSD SBIR with Securboration, 1/19/17 to 1/19/18, $35,000.
9. "A Digital Platform for Social and Emotional Learning," NSF, 7/1/2018 to 12/31/2018, $50,000.
10. "Blockchains Data Exchange via FHIR," Solaster, 9/1/18 to 8/31/19, $30,000, co-PI with Jules White.
11. "Advancing Data-Driven mHealth Technologies for Long-term Health and Health Behavior Change," Trans-Institutional Program (TIPs), Vanderbilt University, 9/1/2016 to 8/31/2018, $100,000, Co-PIs Jules White, Trent Rosenbloom, and Heidi Silver.
12. "IMMoRTALS," DARPA (through subcontract with Raytheon), 12/1/15 to 12/1/19, $1,235,567, Co-PI Jules White.
13. "The Robust Software Modeling Tool (RSMT)," ONR, 7/1/14 to 6/30/17, $749,904, Co-PI Jules White.
14. "Building Resilient Distributed Systems for Next Generation Mobile Adhoc Cyber Physical Systems," Siemens 9/1/14 to 8/31/17, $438,188, co-PI Abhishek Dubey.
15. "Capability-Based Technical Reference Frameworks for Open System Architecture Implementations," OSD ASDR&E, 7/3/14 to 9/11/14, $29,690.
16. "Progressive Model Generation for Adaptive Resilient System Software," ONR STTR, 8/6/13 to 1/31/14, $49,406, co-PI Jules White.
17. "Systems and Software PRodUcibility Collaboration and Experimentation Environment (S2PRUCE2)," AFRL (subcontract through Lockheed Martin Advanced Technology Lab), 1/4/13 to 9/30/13, $108,645, with A. Gokhale.
18. "Stochastic Hybrid Systems Modeling and Middleware-enabled DDDAS for Next-generation US Air Force Systems," AFOSR, 10/1/13 to 9/30/16, $935,402, Co-PI(s) Aniruddha Gokhale and Xenofon Koutsoukous.
19. "Workshop on Computing Clouds for Cyber Physical Systems," NSF, 9/15/12 to 12/31/2013, $73,738.
20. "Using Social Learning to Improve Adolescent Diabetes Protocol Adherence," NIH, $1,798,029, 9/1/12-8/31/16, PI Shelagh Mulvaney.
21. "Systems and Software PRodUcibility Collaboration and Experimentation Environment (S2PRUCE2)," AFRL (subcontract through Lockheed Martin Advanced Technology Lab), 4/3/08 to 9/30/12, $381,708, with A. Gokhale.
22. "Team for Research in Ubiquitous Secure Technology (TRUST)," NSF (subcontract through UC Berkeley), 6/1/05 to 10/31/15, $5,970,900, co-PI(s) J. Sztipanovits and G. Karsai.
23. "Android Mobile Military Middleware Objects (AMMO)," DARPA, 9/30/10 to 5/02/12, $1,074,093, with S. Neema.
24. "Cyber-physical multi-Core Optimization for Resource and cachE effectS (C2ORES)", AFRL, 8/1/12 to 7/31/13, $300,000, with A. Gokhale.
25. "Model-Driven Tools for Distributed- and Multi-Core Middleware," AFRL, 4/10/12 to 10/2/12, $30,000, with A. Gokhale.
26. "Cloud Environmental Analysis and Relief," NSF, 8/1/10 to 7/31/12, $66,000, with A. Gokhale.
27. "Environment-Specific Inter-ORB Protocols," SAIC, 8/1/09 to 5/23/12, $348,350, with A. Gokhale.
28. "CoSMIC and CIAO Enhancements," Northrop Grumman, 7/1/09 to 9/30/10, $878,661
29. "Integrating DDS and CCM," Northrop Grumman, 7/1/09 to 2/15/10, $85,000
30. "Early Integration and Performance Testing of Heterogeneous Computing Environments," Australian Defence Science and Technology Organization (DSTO), 1/9/09 to 7/30/09, $180,000.
31. "Predictive Cache Modeling and Analysis," AFRL (subcontract through Lockheed Martin Aeronautics), 3/1/10 to 9/30/11, $100,000.
32. "Applications of Reliable, Fast Event Notification," Raytheon, 6/1/2008 to 5/30/2009, $60,000.
33. "Open Modular Embedded Architectures," General Electric Global Research, 8/1/2008 to 1/31/2009, $35,000.

34. "Analysis and Simulation Techniques for Next-generation Motion Control Systems," Aagard, 8/1/2008 to 1/31/2009, $13,850 with Akos Ledeczi.
35. "Open Modular Embedded Architectures," Raytheon, 8/1/2008 to 3/31/2009, $74,276.
36. "NAOMI," LMCO Advanced Technology Lab, 9/1/2007 to 11/30/2009, $290,000.
37. "IU/CRC Membership," Siemens, 1/1/2009 to 12/31/2009, $40,000.
38. "Enterprise Application Configuration in the Context of Model Driven Software Development and Software Factories," Siemens Corporate Research, 10/1/07 to 9/31/08 $91,798.
39. "Modular Extendable Demonstration of an Upgradeable Space Architecture (MEDUSA)," DARPA (subcontract through Lockheed Martin Advanced Technology Center), 2/1/2008 to 1/31/2011, $600,000.
40. "CCM Middleware Implementation and Integration," PrismTech, 6/8/2007 to 3/31/2007, $33,778.
41. "The Smart Sensor Web Architecture," NASA (subcontract through Lockheed Martin Advanced Technology Center), 12/15/06 to 11/14/09, $467,728, co-PI G. Biswas.
42. "I/UCRC Membership," General motors, 1/1/2008 to 12/31/2008, $100,000, co-PI G. Karsai.
43. "Pollux: Enhanding the Real-time QoS of the Global Information Grid," AFRL, 2/24/06 to 7/24/08, $1,242,718, co-PI M. Reiter.
44. "Intelligent Middleware for Next Generation Petascale Scientific Computing," Vanderbilt Discover Grant, 5/1/05 to 6/30/07, $100,000, co-PI(s) A. Gokhale and P. Sheldon.
45. "Air Force Center for Research on GIG/NCES Challenges," AFOSR (subcontract through UC Berkeley), 3/1/06 to 2/28/08, $600,000, co-PI J. Sztipanovits.
46. "Quality of Service Enabled Dissemination," AFRL (subcontract through BBN Technologies), 12/31/2007 to 9/30/2009, $320,000.
47. "A Fault-Tolerant Real-Time CORBA Naming Service," US Navy (subcontract through Tech-X Corp), 11/1/2007 to 4/30/2010, $175,000, co-PI A. Gokhale.
48. "System Execution Modeling Technologies for Large-scale Net-centric Systems," AFRL, 1/1/2008 to 12/31/2010, $244,000.
49. "Model-Driven Computing for Distributed Real-time Embedded Systems," Raytheon, 8/31/04 to 8/31/08, $500,000.
50. "NAOMI," LMCO Advanced Technology Lab, 9/1/2007 to 11/30/2007, $50,000.
51. "ACE/TAO Improvement Techniques and Solutions, Veritas/Symantec, 3/31/05 to 4/31/08, $198,500.
52. "Adaptive Resource Control for Certificable Systems," DARPA (subcontract through LMCO Advanced Technology Lab), 3/30/2007 to 12/31/2007, $50,000.
53. "Survivable Internet-scale Distributed Systems," IDA, 3/30/2007 to 12/31/2007, $60,000.
54. "QUality of service pICKER (QUICKER)," LMCO Advanced Technology Lab, 3/30/2007 to 12/31/2007, $60,000.
55. "Thimble," LMCO Advanced Technology Lab, 3/30/2007 to 12/31/2007, $60,000.
56. "CADynCE Experimentation Operations (CEO)," DARPA (subcontract through LMCO Advanced Technology Lab), 8/31/2007 to 12/31/2007, $25,000.
57. "Real-time Discovery for Pub/Sub Middleware in WANs," US Navy (subcontract through Tech-X Corp), 6/16/2007 to 9/31/2007, $15,000.
58. "GEMS Utilization Test Suite," LMCO Advanced Technology Lab, 9/1/07 to 11/30/07, $50,000.
59. "Advanced Information Systems and Technology Program," NASA (subcontract through LMCO Advanced Technology Center), 11/13/2007 to 12/1/2007, $22,000, co-PI G. Biswas.
60. "Design for Adaptivity and Reliable Operation of Software Intensive Systems," NSF CNS-0613971, 9/1/06 to 8/31/08, $199,867, co-PI(s) S. Abdelwahed and G. Karsai.
61. "Software Technologies Targeting Interoperability for Systems of Systems," Army Research Lab, 1/15/07 1/14/10, $851,567, co-PI(s) G. Karsai and J. Sztpanovits.
62. "Software Wind Tunnel (SWiT) Capabilities," Lockheed Martin Advanced Technology Lab, 8/1/06 to 12/31/06, $60,000.

63. “High-Confidence Software Platforms for Cyber-Physical Systems,” NSF, 5/1/06 to 7/30/08, $129,179.
64. “Applying AOP to Develop of Component Synthesis with MDD,” Siemens, 3/1/03 to 2/28/07, $400,005.
65. “Addressing Domain Evolution Challenges in Model-Driven Software Product-lines,” Siemens Corporate Research, 10/1/05 9/31/07, $100,000.
66. “A Fault Tolerant Real-time CORBA Naming Service,” US Navy (subcontract through Tech-X Corp), 11/1/05 to 8/31/06, $15,000.
67. “The SYstem DEployment and Configuration AssisteR (SYDECAR),” Lockheed Martin Advanced Technology Lab, 8/1/05 to 8/1/08, $500,000.
68. “Future Combat Systems: Software Architecture Engineering,” DARPA (subcontract through Boeing), 1/28/05 to 12/31/07, $2,764,226, co-PI(s) J. Sztipanovits and G. Karsai.
69. “Development of an Eclipse Plug-in,” PrismTech, 4/28/05 to 9/30/05, $25,000.
70. “Prometheus: Enhancing the QoS of the JBI,” AFRL, 3/25/05 to 12/31/05, $500,000, co-PI(s) K. Birman and Mike Reiter.
71. “A Testbed for Assuring Quality of Software for DRE Systems,” ONR, 2/15/05 to 1/31/06, $200,000, co-PI(s) A. Gokhale and A. Porter.
72. “Enhancing the QoS of SOAs Using Eclipse-based MDD,” IBM, 2/15/05 to 1/31/06, $29,515, co-PI A. Gokhale.
73. “Model-Driven Development of BEEP Application Protocols,” Cisco, 12/15/04 to 12/14/05, $57,976, co-PI A. Gokhale.
74. “Evaluating CORBA Middleware for Space Systems,” NASA (subcontract through Lockheed Martin Advanced Technology Center), 9/23/04 to 11/30/06, $186,180, co-PI G. Biswas.
75. “Refactoring Techniques to Reduce Middleware Resource Utilization,” Qualcomm, 10/31/04 to 10/31/05, $104,000, co-P B. Natarajan.
76. “Model-Driven Development for Software Defined Radios,” BAE Systems, 12/1/04 to 3/31/05, $32,000.
77. “Enhancing the Robustness and Performance of TENA,” DISA (subcontract through SAIC and OSC), 7/1/04 to 12/31/04, $75,000.
78. “QoS-enabled Fault Tolerant Middleware and MDA Tools,” Lockheed Martin MSS, 4/1/03 to 12/31/04, $516,434.
79. “Trustworthiness in Embedded Systems,” NSF ITR CCR-032574, 9/31/03 to 8/31/06, $210,454.
80. “ACE+TAO Enhancements,” OCI, gift $20,000.
81. “Acquiring Accurate Dynamic Field Data Using Lightweight Instrumentation,” NSF ITR CCR-0312859, 10/1/02 to 9/31/07, $1,850,000, co-PI(s) A. Porter, D. Notkin, and A. Karr.
82. “Intergovernmental Personnel Act,” DARPA, 6/1/00 to 5/31/02, $198,934.
83. “Optimizing Component Models,” DARPA, 4/1/01 to 6/31/02, $210,000.
84. “HLA RTI Next-generation,” DMSO (subcontract through SAIC), 6/1/01 to 12/31/01, $70,895.
85. “ACE Enhancements for Windows NT and Windows CE,” Siemens Medical Engineering, 2/1/00 9/19/01, $112,000.
86. “Scalable and Fault Tolerant Middleware,” AFRL MURI, 12/1/99 to 3/31/02, $253,701.
87. “Protocol Engineering Research Center,” AFOSR MURI, 6/15/00 to 6/14/03, $264,720, co-PI Tatsuya Suda.
88. “Optimizing ORBs for Network Management,” Cisco Systems, 1/1/00 to 12/31/00, $100,000.
89. “TAO Optimizations,” Raytheon, 10/1/99 to 6/01/01, $50,000.
90. “ACE+TAO on pSoS,” Motorola, 8/15/99 to 12/31/99, $30,000.
91. “Real-time Distributed Object Computing,” Sprint, 8/15/99 8/14/00, $133,068.
92. “TAO Enhancements,” Krones, 8/1/99 to 9/1/99, $5,000.
93. “ACE Enhancements,” ICOMVERSE, gift, $20,000.

94. “Weapon Systems Open Architecture,” Boeing, 7/15/99 to 1/31/00, $51,491.
95. “Fault Tolerant CORBA,” Motorola Labs, 7/15/99 to 7/14/00, $139,000.
96. “TAO Enhancements,” Global MAINTECH, 7/1/99 to 8/1/99, $5,000.
97. “ACE QoS Extensions,” Motorola Trunking, 6/1/99 to 8/1/99, $5,000.
98. “CORBA Interceptors,” Experian, 5/15/99 7/14/99, $10,000.
99. “DCOM performance evaluation,” Microsoft, gift, $30,000.
100. “TAO Improvements,” OCI, 4/1/99 to 9/31/00, $27,000.
101. “Middleware Optimizations,” Telcordia, 2/1/99 to 1/31/00, $52,700.
102. “Minimum CORBA,” Hughes Data Networking, 4/1/99 to 3/31/00, $50,000, co-PI David Levine.
103. “Framework Usage Patterns,” Siemens Corporate Research, 4/1/99 to 3/31/00, $35,000.
104. “Dynamic Scheduling and Real-time ORB Optimizations,” Boeing, 10/1/98 9/30/99, $184,860.
105. “Distributed Object Computing Middleware,” Nortel, 11/1/98 10/31/99, $75,000.
106. “ACE subsetting,” “ACE subsetting,,” Nokia, 10/8/98 4/8/99, $30,000.
107. “Boeing Research Fellowship,” Boeing, 9/1/98 8/31/00, $81,486.
108. “Patterns and Frameworks Reuse Curriculum,” Lucent Bell Labs, 9/1/98 12/31/98, $31,200.
109. “Patterns, Frameworks, and Components,” Siemens ZT, 12/1/98 5/31/00, $175,000.
110. “High availability frameworks,” Lucent, 9/1/98 8/31/99, $39,400.
111. “Real-time Distributed Object Computing,” Sprint, 8/1/98 7/31/99, $288,194.
112. “Distributed Object Integration for the Quorum Project,” DARPA S30602-98-C-0187 (subcontract through BBN), 9/1/98 8/31/01, $448,643, co-PI(s) R. Schantz and J. Loyall.
113. “Evaluating a Framework for Dynamic Distributed Real-Time Scheduling,,” USENIX, gift, $18,000.
114. “Distributed Object Computing,” Microsoft, gift, $20,000.
115. “Distributed Object Visualization Environment,” Lockheed Martin, 5/1/98 to 11/31/99, $54,000.
116. “Distributed Object Computing with Adaptive End-to-end QoS Guarantees,” DARPA 9701561, 8/1/97 to 7/31/00, $873,625.
117. “Real-time CORBA for Telecommunications,” Lucent, 12/1/97 to 11/31/98, $100,000.
118. “Developing an HLA-compliant RTI with ACE,” SAIC, 12/15/97 to 1/31/00, $228,075.
119. “Real-time CORBA for Wireless,” Motorola LMPS, 10/15/97 to 10/14/98, $200,000.
120. “Real-time CORBA for Avionics,” Computing Devices International, 10/15/97 to 10/14/98, $39,050.
121. “Dynamic Scheduling of Real-time OFPs,” Boeing, 9/1/97 to 8/31/98, $224,604.
122. “Distributed Object Visualization,” Siemens MED, 10/1/97 to 9/1/98, $40,000.
123. “The ADAPTIVE Communication Environment,” Siemens MED, 10/1/97 to 9/1/98, $70,000.
124. “The Architect’s Assistant,” Siemens Corporate Research, 9/1/97 to 8/1/98, $35,000.
125. “Monitoring, Visualization, and Control of High Speed Networks,” NSF NCR-97-14698, 9/1/97 to 8/31/01, $1,200,000, co-PI(s) G. Parulkar, E. Kraemer, J. Turner, and R. Cytron .
126. “Adaptive Software Technology Demonstration (ASTD),” AFRL (subcontract through Boeing), 9/1/98 to 8/31/02, $1,200,000, co-PI(s) B. Doerr, D. Allen, and R. Jha.
127. “Patterns, Frameworks, and Components for Multimedia Systems,” Siemens Research, 1/97 to 6/98, $150,000.
128. “Adaptive Servers for High-Performance Imaging,” Kodak Networked Imaging Tech. Center, 11/96 to 11/97, $40,000.
129. “Real-time CORBA,” Sprint, 9/96 to 12/97, $345,000, co-PI G. Parulkar.
130. “OpenMAP – Object-Oriented Components for Real-time Avionics,” McDonnell Douglas, 9/96 to 9/97, $241,591.
131. “Compilation and Automatic Optimization of Network Protocol Implementations,” NSF NCR-9628218, 8/96 to 8/99, $411,025, co-PI(s) G. Varghese and R. Cytron (PI).

132. "Medical Imaging with Java and the WWW," SIEMENS Medical Engineering, 8/96 to 7/97, $125,000.
133. "The ADAPTIVE Communication Environment," SIEMENS Medical Engineering, 8/96 to 7/97, $90,000.
134. "High-performance Distributed Medical Imaging," Kodak Imaging, 12/94 to 8/96, $55,152, co-PI J. Blaine.
135. "Design Patterns for Concurrent Object-Oriented Networking," Object Technologies International, 4/96 to 4/97, $25,000.
136. "Distributed Object Computing with CORBA and DCE," Bellcore, 5/96 to 12/96, $32,978.
137. "The ADAPTIVE Communication Environment," SIEMENS Medical Engineering, 6/95 to 6/96, $170,000.

## Courses Taught

Courses at Vanderbilt University

1. CS 215 – Intermediate Software Design, Spring 2006
2. CS 251 – Intermediate Software Design, Spring 2007, Spring 2008, Spring 2009, Fall 2009, Spring 2010, Spring 2012, Spring 2013, Spring 2014, Spring 2015, Spring 2016
3. CS 291/242 – Software Design Studio, Fall 2004
4. CS 291/242 – Software Design Studio, Fall 2003
5. CS 292 – Beyond the Oneway Web, Fall 2008
6. CS 278 – Software Engineering, Fall 2008
7. CS 279 – Software Engineering Projects, Spring 2010
8. CS 282 – Principles of Operating Systems II, Spring 2003, Spring 2004, Fall 2005, Fall 2007, Fall 2012, Fall 2013, Fall 2014, Fall 2015, Fall 2016, Spring 2017
9. UNIV 278 – Tackling Big Questions with Mobile Cloud Computing, Fall 2016, Spring 2017, Fall 2017
10. CS 395 – Advanced Network Software Design, Fall 2006
11. CS 395 – QoS-enabled Middleware, Fall 2008
12. CS 396 – QoS-enabled Component Middleware, Spring 2005
13. CS 891 – Introduction to Concurrent and Parallel Java Programming with Android, Fall 2017
14. CS 891 – Advanced Concurrent Java Programming in Android, Spring 2018
15. CS 891 – Introduction to Parallel Java Programming, Fall 2018
16. CS 892 – Concurrent Java Programming in Android, Spring 2017

Courses at Coursera

1. Android App Development (Android for Java; Android App Components - Intents, Activities, and Broadcast Receivers; Android App Components - Services, Local IPC, and Content Providers), 2016 to present
2. Mobile Cloud Computing with Android (Pattern-Oriented Software Architecture: Communication; Pattern-Oriented Software Architecture: Concurrency), 2014 to 2016
3. Pattern-Oriented Software Architectures for Concurrent and Networked Software, 2013

Courses at University of California, Irvine

1. ECE 011 – Computational Methods in ECE, Winter 2000
2. ECE 255 – Distributed Software Architecture Design, Spring 2000
3. ICS 142 – Compiler Theory, Summer 1989
4. ICS 23 – Data Structures, Summer 1988

Courses at Washington University, St. Louis

1. CS 562 – Advanced Object-Oriented Software Development with Patterns and Frameworks, Spring 1999
2. CS 242 – Introduction to Software Design, Spring 1998
3. CS 673 – Distributed Systems research seminar, Fall 1997
4. CS 422 – Operating Systems Organization, Fall 1997
5. CS 242 – Introduction to Software Design, Spring 1997
6. CS 544 – Distributed System Design, Fall 1996
7. Ada tasking course for McDonnell Douglas, Fall 1996
8. OO design course for McDonnell Douglas, Spring 1996
9. CS 523 – Distributed Operating Systems Organization, Spring 1995
10. CS 242 – Introduction to Software Design, Fall 1995
11. CS 673 – Distributed Systems research seminar, Spring 1995
12. CS 422 – Operating Systems Organization, Fall 1994

**Other Teaching Experience**

In addition to the academic teaching experience above, I have also taught numerous short-courses and tutorials on object-oriented design patterns and programming techniques, UNIX and Windows NT systems programming and network programming, C++ and C programming languages, and various distributed/networked system, compiler construction, algorithm, data structure, mobile app, and web-based cloud computing courses for the following universities and professional organizations:

- O'Reilly Live-Training
- Pearson LiveLessons
- University Extension Program, University of California, Berkeley, CA
- University Extension Program, University of California, Irvine, CA
- University Extension Program, University of California, Los Angeles, CA
- Oregon Graduate Institute of Science and Technology, Beaverton, OR
- USENIX association
- Association of Computing Machinery (ACM)
- Addison-Wesley's Technology Exchange Program, Reading, MA
- SIGS Conferences
- Object Computing Institute, St. Louis, MO
- National University, Irvine, CA

## Department/School/Community Service

Service at Vanderbilt University

1. Faculty advisor for the "DataBrains" AI and Data Science student club.
2. Faculty advisor for the "Vandy Apps" student club.
3. Faculty advisor for the "BizTech" student club.
4. Led the effort to create an online Professional Masters in CS
5. Led the effort to create a continuing education program in Web Development
6. Interview panel for the Director of Professional Programs in VUSE
7. Served on the Digital Literacy committee
8. Chair of two year review committee for Taylor Johnson
9. Chair of the CS search committee in 2003, 2005, 2013, 2016, 2018
10. Chair of the committee on Big Data for the VUSE Strategic Plan
11. Member of the Provost's Special Task Force of the Data Science Visions Working Group: Trans-institutional Masters in Data Science.

12. Member of the Provost's Data Science Visions working group
13. VUSE representative for the Research IT committee
14. VUSE representative on the Provost's Digital Literacy committee
15. Reviewer for University Course proposals
16. Faculty mentor for "Accenture Garage Program"
17. VUSE representative for the Research IT committee.
18. Member of the search committee for the first Director of the Innovation Center
19. Member of the Provost's Study Group on Cross College Teaching
20. Member of the Advisory Committee for the Vanderbilt Institute for Digital Learning (VIDL)
21. Chair of the Provost's Committee on the Innovation Center
22. Member of the VUSE Career Committee
23. VUSE point of contact for VUIT
24. Committee member for Eugene Vorobeychik's promotion case to associate professor
25. Committee member for Bobby Bodenheimer's promotion case to full professor
26. Committee member for Julie Adams's promotion case to full professor
27. Committee member for Akos Ledeczi's promotion case to full professor
28. Chair of the tenure committee for Yuan Xue
29. Chair of the four year review committee for Yuan Xue
30. Member of the two year committee for Yuan Xue
31. Member of the promotion committee for Ted Bapty
32. Member of review committee for Xenofon Koutsoukos
33. Chair of promotion committee for Gabor Karsai
34. Member of promotion committee for Gautam Biswas
35. Chair of the VUSE Technology Entrepreneurship Task Force
36. Member of the VUIT faculty advisory committee
37. Owen-VUSE joint committee for 2014-2015
38. Chair of the Schmidt Family Annual Educational Technologies Lectureship
39. Member of the Provost's Study Group on Cross College Teaching
40. Chair of two year review committee for Eugene Vorobeychik
41. Member of the Chancellor's Social Media and the Internet committee
42. Member of the VU Online Education Task Force
43. Member of the ad hoc committee on EECS Industrial Advisory Board
44. Ex-officio member of the ad hoc committee on the CS graduate program
45. Ex-officio member of the ad hoc committee on the CS undergraduate program
46. Faculty facilitator for the Vanderbilt Visions program
47. Chair of the Information Technology committee for the Vanderbilt School of Engineering
48. Chair of the tenure committee for Bobby Bodenheimer
49. EECS Corporate/Internship Liaison for Computer Science and Engineering
50. Ex-officio Member of the Ad Hoc Committee on Computer Engineering
51. Faculty sponsor of the new EECS Graduate Student Organization
52. Member of the VUSE Research Institutes and Centers Council
53. Associate Chair of Computer Science and Engineering
54. Member of the Vanderbilt University Faculty Senate
55. Chair of the faculty committee on Academic Computing and Information Technology (ACIT)

56. Member of the Research Advisory Committee on Information Technology (RACIT)
57. Chair of the Systems Engineering concentration committee
58. Member of the Plan Integration and Communication Group (PICG)
59. Member of the CS graduate curriculum committee

Service at Washington University, St. Louis

1. Member of the Faculty recruiting committee
2. Member of the CS committee on recruiting industrial graduate students (RIGS)
3. Member of the CS Experimental Infrastructure for Teaching and Research (CEITR)
4. Member of the Introductory course committee
5. Member of the Graduate admission committee
6. Member of the CS representative to the CEC advisory board
7. Member of CS departmental chair search committee

## Awards and Honors

1. Received the Cornelius Vanderbilt Professor of Engineering endowed chair in February 2017.
2. Received the 2015 Award for Excellence in Teaching by the Vanderbilt University School of Engineering.
3. Interviewed for Software Engineering Radio (www.se-radio.net/).
4. Vice-chair of the IEEE Chapter in middle Tennessee.
5. Elected to three year term as member of the Vanderbilt University Faculty Senate.
6. Invited speaker at the dedication of the Henry Samueli School of Engineering, along with UC Irvine Chancellor, Ralph Cicerone; Dean of the School of Engineering, Nicolaos Alexopoulos; Chairperson of the Regents of the University of California, S. Sue Johnson; President of the University of California, Dick Atkinson; and CTO and co-founder of Broadcom Henry Samueli.
7. Interviewed for Dr. Dobb's journal TechNetCast, October 24, 2000.
8. Interviewed for **iX** magazine, October, 2000.
9. Received early promotion to tenure as an Associated Professor at Washington University, St. Louis, five years after joining the faculty as an Assistant Professor in 1994.
10. Director of the "Center for Distributed Object Computing" at Washington University, St. Louis since spring of 1999.
11. Listed in Marquis' "Who's Who in Media and Communications," 1997.
12. Received joint appointment to the Mallinckrodt Institute Department of Radiology, Washington University School of Medicine, February 1996.
13. Selected to participate in the ACM OOPSLA '94 Doctoral Symposium.
14. Invited by Dr. Martina Zitterbart to participate in a 4-week international exchange program at the Universität Karlsruhe Institut für Telematik in Karlsruhe, Germany, April 1993.
15. Served as elected representative to the Associated Graduate Student organization at the University of California, Irvine from May 1991 to June 1992.
16. Served as elected graduate student representative to the Computer Science Computing Resource Committee at the University of California, Irvine from August 1988 to August 1990.

## Consulting Work

1. ARINC, Fountain Valley, CA
2. ACM, NY, NY
3. Advanced Institute of Information Technology, Seoul, Korea
4. AG Communication Systems, Phoenix, AZ
5. Anderson Consulting, Chicago, IL
6. Apple, Cupertino, CA
7. AT&T Research, Murray Hill, NJ
8. BAE Systems, Greenlawn, NY
9. BAE Systems, Wayne, NJ
10. BEA, San Jose, CA
11. Bellcore, Morristown, NJ
12. BellSouth, Atlanta, GA
13. Boeing, St. Louis, MO
14. Boies, Schiller, & Flexner, Santa Monica, CA
15. Bridges & Mavrakakis, Palo Alto, CA
16. Cooley LLP, San Francisco, CA
17. Correct Care Solutions, Nashville, TN
18. Credit Suisse, Zurich, Switzerland
19. Crosskeys, Ottawa, Canada
20. DARPA, Arlington, VA
21. Desmarais, NY, NY
22. Duane Morris, Atlanta, GA
23. Edward D. Jones, St. Louis, MO
24. Envision Inc. St. Louis, MO
25. Ericsson, Cypress, CA
26. Fitzpatrick, Cella, Harper & Scinto, NY, NY
27. GaN Corporation, Huntsville, AL
28. Gibson, Dunn, & Crutcher, NY, NY
29. Goldman, Ismail, Tomaselli, Brennan, & Baum, Chicago, IL
30. Jet Propulsion Lab, Pasadena, CA
31. Kasowitz, Benson, & Torres, Redwood Shores, CA
32. Keystone Strategy, Boston, MA
33. Kilpatrick Stockton, Atlanta, GA
34. Kirkland & Ellis, San Francisco, CA
35. Kodak Imaging, Rochester, NY
36. Laureate University, Baltimore, MD
37. Lockheed Martin Tactical Systems, Minneapolis, MN
38. Lockheed Martin Mission Systems, Boulder, CO
39. Lockheed Martin Advanced Technology Lab, Cherry Hill, NJ
40. Lucent Bell Labs, Naperville, IL
41. Lucent Bell Labs, Murray Hill, NJ
42. Lucent, Whippany, NJ

43. McDonnell Douglas, St. Louis, MO
44. Microsoft, Redmond, WA
45. Morrison & Foerster, Washington DC
46. Morgan Stanley, New York, NY
47. Motorola Cellular Infrastructure Group, Arlington Heights, IL
48. Motorola Iridium, Chandler, AZ
49. Motorola Land Mobile Products, Chicago, IL
50. National Security Agency, Ft. Meade, MD
51. Naval Air Weapons Stations, China Lake, CA
52. Nortel, Ottawa, Canada
53. Object Computing Institute, St. Louis, MO
54. Object Technologies International, Ottawa, CA
55. Odetics Broadcasting, Anaheim, CA
56. Oracle, Redwood Shores, CA
57. Park, Vaughan, & Fleming, Boise, ID
58. Pearson Education, London, UK
59. Pragmatus, Alexandria VA
60. PrismTechnologies, Newcastle, UK
61. Qualcomm, San Diego, CA
62. Quinn Emanuel, NY, NY
63. Raytheon, San Diego, CA
64. Reichman Jorgensen, CA
65. Riverace, Boston, MA
66. Rubin Anders Scientific, Boston, MA
67. SAIC, Washington D.C.
68. Schwegman, Lundbert, & Woessner, Minneapolis, MN
69. Siemens Medical Engineering, Erlangen, Germany
70. Siemens Corporate Research, Princeton, NJ
71. SIGS, New York, NY
72. Software Engineering Institute, Pittsburgh, PA
73. Teradyne, Chicago, IL
74. Teledyne, Thousand Oaks, CA
75. UC Berkeley Extension, Palo Alto, CA
76. UCLA Extension, Los Angeles, CA
77. USENIX, Lake Forest, CA
78. Venable, NY, NY
79. Wong, Cabello, Lutsch, Rutherford & Brucculeri, Houston, TX
80. WMS Gaming, Chicago, IL
81. Zircon Computing, Wayne, NJ

## Expert Testimony in the Past Five Years

1. March 2016, Deposed in support of Oracle in the Oracle vs. Google Fair Use trial in the United States District Court for the Northern District of California, San Francisco division. Case No. Civ. A. No. 10-03561 WHA.

2. May 2016, Testified in support of Oracle in the Oracle vs. Google Fair Use trial in the United States District Court for the Northern District of California, San Francisco division. Case No. Civ. A. No. 10-03561 WHA.

3. February 2017, Deposed in support of IBM in the IBM vs. Priceline Group case. Case No. Civ. A. N. 15-cv-137-LPS-CJB.

4. February 2018, Deposed in support of IBM in the IBM vs. Groupon case. Case No. Civ A. N. 16-122-LPS-CJB.

5. July 2018, Testified in support of IBM in the IBM vs. Groupon case. Case No. Civ A. N. 16-122-LPS-CJB.

6. August 2018, Deposed in support of Palo Alto Networks in the Palo Alto Networks vs. Implicit case. Case No. Civ 6:17-CV-182-JRG.

7. January 2019, Deposed in support of C3IoT in the E2.0 vs. C3IoT case. Case No. 1:15-cv-00530-GMS.

8. February 2019, Testified in support of C3IoT in the E2.0 vs. C3IoT case. Case No. 1:15-cv-00530-GMS.

9. June 2019, Deposed in support of IBM in the IBM vs. Expedia Inc. case. Civil Action No. IPR2018-01136.

10. July 2019, Deposed in support of Philips in the Philips vs. Microsoft case. Civil Action No. 4:18-cv-01885-HSG.

11. August 2019, Deposed in support of Philips in the Philips vs. HTC case. Civil Action No. 4:18-cv-01885-HSG.

12. August 2019, Deposed in support of Philips in the Philips vs. ASUS case. Civil Action No. 4:18-cv-01885-HSG.

13. September 2019, Deposed in support of Kroy in the Kroy vs. Groupon case. Civil Action No. IPR2019-00044.

14. September 2019, Deposed in support of Kroy in the Kroy vs. Groupon case. Civil Action No. IPR2019-00061.

15. March 2020, Deposed in support of Cisco in the Centriptal vs. Cisco case. Civil Action No. 2:18-cv-00094-HCM-LRL.

16. May 2020, Testified in support of Cisco in the Centriptal vs. Cisco case. Civil Action No. 2:18-cv-00094-HCM-LRL.

17. Jan 2021, Deposed in support of Droplets in the Droplets vs. Yahoo case. Civil Action No. 12-CV-03733-JST.

18. Jan 2021, Deposed in support of Droplets in the Droplets vs. Nordstrom case. Civil Action No. 12-CV-04049.

## Summary of Research Contributions

At Vanderbilt University I direct the Distributed Object Computing (DOC) Group at the Institute for Software Integrated Systems (ISIS), which is one of the leading research groups in the world on middleware platforms and MDE tools for DRE systems and mobile cloud computing platforms. Over the past several decades I have conducted and managed research projects on a range of topics, including patterns, optimization techniques, and empirical analyses of software frameworks that facilitate the development of quality of service (QoS)-enabled middleware and model-driven engineering (MDE) techniques/tools for distributed real-time and embedded (DRE) systems and mobile cloud computing apps running over wired/wireless networks and embedded system interconnects. The research methodology throughout my career has involved:

- *Creating* innovative middleware and MDE technologies technologies, such as design formalisms, QoS specification/enforcement techniques, end-to-end and cross-layer middleware optimizations, and automated tools for specifying, analyzing, and synthesizing dependable DRE software from higher-level domain-specific models.
- *Applying* these technologies in conjunction with colleagues in academia and industry to demonstrate and mature middleware and MDE technologies and tools in the context of production mission-critical DRE systems.
- *Amplifying* the adoption and transition of these technologies in both academia and industry via 625+ technical papers, 575+ tutorials and invited talks, millions of lines of popular open-source software, and scores of innovative face-to-face and online courses published and delivered to more than 300,000 students around the world.

The R&D efforts I have led have had a significant impact on academic research and commercial practice. For example, dozens of universities throughout the world use the middleware and MDE tools my DOC Group has developed as the basis for their research and teaching efforts. Moreover, the open-source middleware frameworks and MDE tools generated from projects I've led constitute some of the most successful examples of software R&D ever transitioned from research to industry, being widely used by thousands of companies and agencies worldwide in many domains for three decades. For example, the ACE and TAO middleware frameworks developed by the DOC Group are used by developers in thousands of companies (such as Boeing, Cisco, Ericsson, Kodak, Lockheed Martin, Lucent, Motorola, NASA/JPL, Nokia, Nortel, Raytheon, SAIC, Siemens, Sprint, and Telcordia) in a wide range of domains (such as telecom/datacom, healthcare, process automation, avionics, homeland security and defense, financial services, online gaming, social media, and distributed interactive simulation).

### Teaching Contributions and Impact

I have taught scores of cutting-edge courses on topics relating to object-oriented design and programming, software patterns, middleware for distributed real-time and embedded systems, concurrent and networked programming with C++ and Java, and mobile cloud computing with Android. I received the 2015 Award for Excellence in Teaching by the Vanderbilt University School of Engineering. In addition, I've taught 10 popular MOOCs at Vanderbilt on topics related to pattern-oriented mobile cloud computing with Android to over 200,000 learners from around the world.

I recently created and co-taught one of the first cross-college University Courses at Vanderbilt on "Tackling Big Problems with Mobile Cloud Computing," where ten highly diverse teams consisting of 11 arts and science students and 44 computer science students were mentored by 11 faculty from the College of Arts and Sciences, the School of Nursing, the School of Law, the School of Medicine, the School of Engineering and Vanderbilt University Medical Center. The projects in this course addressed relevant, real-world problems involving mobile cloud computing technologies, including:

- Effectively engaging young people with chronic diseases and medical conditions, such as diabetes, asthma and obesity
- Creating "smarter" cities and sustainable energy platforms via an app-based transportation hub for Nashville, and remotely monitoring the safety and operations of novel sources of power, including solar, wind and natural gas, and
- Helping economically disadvantaged individuals bridge the digital divide to obtain better guidance on medical and legal matters.

## Summary of Career Accomplishments

My career accomplishments include the following:

**Publications and presentations.** I have published 650+ works (127 journal papers, 195 conference papers, 5 books, 4 book-length reports, 3 edited book collections, 64 book chapters, 74 workshop papers, 13 short papers and posters, 75 trade magazine columns/articles, and 101 editorials and book forewords). My papers have appeared in the most selective journals (*e.g.*, ACM Transactions in Embedded Computing Systems, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Software Engineering, IEEE Transactions on Computing, IEEE Journal of Selected Areas of Communications, and ACM Transactions on Autonomous and Adaptive Systems) and conferences (*e.g.*, ACM SIGCOMM, ACM OOPSLA, IEEE INFOCOM, IEEE ICDCS, IEEE RTAS, ACM/IEEE Middleware, and the ACM/IEEE ICSE) in my field. I have also given 600+ invited lectures and tutorials world-wide.

**Measures of scholarly impact.** My publications have been cited 36,000+ times across a comprehensive spectrum of high-impact venues. My h-index is 86 and my i10 index is 394. These biometrics indicate the significant impact of my publications as a researcher in the field of Computing.

**Funding.** Since June 1995 I have been a PI or co-PI for grants, contracts, and gifts totaling more than $41 million dollars. I have been the sole PI for over $11.5 million dollars of this amount.

**Graduate advising and training.** During my academic career I have advised and graduated 18 doctoral students and over 25 masters students.

**Professional service and leadership.** I have engaged in the following professional service and leadership capacities during my career:

- Served as guest editor of 12 ACM, IEEE, and USENIX journals, and served as editor-in-chief of the C++ Report magazine.
- Served as general chair or program (co)-chair for 35 conferences, tutorial chair for 4 conferences, co-organized 14 workshops, and served on the program committees for over 245 ACM, IEEE, IFIP, USENIX, and OMG conferences.
- From 2013 to 2015 I served on the Advisory Board for the joint US Navy/Army Future Airborne Capability Environment (FACE).
- From 2013 to 2015 I served as co-lead of a task area on "Published Open Interfaces and Standards" for the US Navy's Open Systems Architecture initiative.
- From 2010 to 2014 I served a member of the Air Force Scientific Advisory Board, where I was the Vice Chair of a study on Cyber Situational Awareness for Air Force mission operations.
- From 2006 to 2011 I served as the Chief Technology Officer for the Software Engineering Institute at Carnegie Mellon University (2010 to 2011), Zircon Computing (2009 to 2010), and Prism Technologies (2006-2008), where I was responsible for directing the technical vision and strategic R&D investments.
- From 2000 to 2003 I served as a Program Manager at the DARPA Information Technology Office (ITO) and Information eXploitation Office (IXO) the Deputy Director for DARPA ITO, where I lead the national R&D effort on QoS-enabled middleware for DRE systems.
- From 2001 to 2003 I served as Co-chair for the Software Design and Productivity (SDP) Coordinating Group, which formulates the multi-agency research agenda in fundamental software design for the Federal government's Information Technology Research and Development (IT R&D) Program, which is the collaborative IT research effort of the major Federal science and technology agencies.

**University service and leadership.** I have engaged in the following service and leadership capacities at Vanderbilt University during the past two decades:

- **Associate Provost of Research**. I became the Associate Provost for Research at Vanderbilt University in July of 2018. In this capacity I am responsible for developing cohesive and sustainable information technology (IT) services to advance research and scholarship across Vanderbilt's ten schools and colleges, including scalable and secure storage, processing, and communication solutions; big data research cores and corerelated services, and NIST 800-171 compliant IT services. I am also responsible for overseeing Vanderbilt's new "liquid workforce" service that provides researchers with on-demand access to shared technology expertise to help them develop research IT solutions, especially with data-intensive workflows, while also enabling shared software developers to add value to multiple research programs throughout the university.

- **Data Sciences Initiatives**. I am deeply involved in Vanderbilt's initiatives on Data Science. Starting in August 2018, I became a founding Co-Director of the Data Science Institute at Vanderbilt. During the past year I also chaired the ad hoc committee on Big Data for the Vanderbilt University School of Engineering (VUSE) strategic planning process, as well as served on the Provost's Special Task Force on a trans-institutional Masters in Data Science and the Provost's Working Group on Data Science Visions, which sets the direction for trans-institutional Data Science research. I also created and led a presentation on "Big Data" for the Vanderbilt University Board of Trust in the spring of 2017 that helped initiate Vanderbilt's investment in the Data Science Institute.
- **Cross-College Teaching**. I am a leader in Vanderbilt University's forays into Cross-College teaching. For example, I served as a member of the Provost's Study Group on Cross College Teaching, which formulated the concept of "University Courses" that brings faculty together from multiple schools to actively engage students of diverse backgrounds and promote new and creative trans-institutional learning. I also created/taught one of the first University Courses on "Tackling Big Problems with Mobile Cloud Computing." Each semester since the fall of 2016 I've taught this course in a multidisciplinary environment where undergraduate and graduate students from multiple schools team with computer science students to address big questions, such as how mobile cloud computing technologies can engage young people with chronic diseases; change political discourse in the United States and around the world; and help economically disadvantaged individuals bridge the digital divide to obtain better guidance on nutrition and legal matters. I also spearheaded the effort to create a CS 1000 course on "the beauty and joy of computing" that is intended for non-CS majors at Vanderbilt University.
- **Digital Learning**. I play a significant role in Vanderbilt's digital learning initiatives, including teaching (1) the first Massive Open Online Course (MOOC) at Vanderbilt in 2013 on "Pattern-Oriented Software Architecture for Concurrent and Networked Systems," (2) the first trans-institutional MOOC Specialization (together with the University of Maryland, College Park) in 2014 on "Mobile Cloud Computing with Android," (3) a Coursera Specialization on "Android App Development" since the spring of 2016, and (4) the forthcoming online Computer Science professional master's degree being created in conjunction with 2U. I have also played a key role in formulating the Vanderbilt digital learning strategy as a member of the Advisory Committee for the Vanderbilt Institute for Digital Learning (VIDL), a member of the Vanderbilt Online Education Task Force, a member of the Chancellor's Social Media and the Internet committee, chair of the Schmidt Family Annual Educational Technologies Lectureship, and a member of the Provost's committee on Digital Literacy whose charter is to ensure that all Vanderbilt students learn computational thinking in their undergraduate experience.
- **Technology Entrepreneurship**. I have been highly engaged in entrepreneurship leadership at Vanderbilt over the past five years. In particular, I chaired the VUSE Technology Entrepreneurship Task Force and the Provost's Committee on the Vanderbilt Innovation Center, known as the Wond'ry (I also served as a member of the search committee for the first Director of the Wond'ry Innovation Center). I am one of the inaugural faculty mentors for the "Garage Program at the Wond'ry, where I mentor multi-disciplinary teams of undergraduate and graduate students to help companies (such as Accenture and RGP) establish new lines of business, e.g., liquid workforce services for the oil and gas domain, supply chain risk management using blockchain technologies, etc. I also serve as the faculty advisor for the VandyApp, DataBrains, and BizTech student organizations, which teach software development skills, prepare students for technical job interviews, and foster a welcoming and diverse environment for high-tech entrepreneurship collaboration across campus.
- **EECS Department Leadership**. I served as the Associate Chair of the Electrical Engineering and Computer Science (EECS) department at Vanderbilt University from 2004 to July 2018. In this capacity I worked with the EECS Chair to provide intellectual leadership and assist in EE, CS, and CompE faculty hiring, curricular development, and course staffing. I also represented Vanderbilt at the bi-annual CRA "CS Chairs" meeting at Snowbird Utah since 2008. In the past several years I focused on innovative digital learning techniques (such as pre-recording material and/or recording lectures in class so students can listen/watch to them at their leisure to ensure they master the course material) to handle the surge in undergraduate CS enrollment without adversely affecting Vanderbilt's commitment to high quality education. I also spearheaded several initiatives to create a continuing education program focused on web development in partnership

with Trilogy Education Services and a professional masters degree program in CS in conjunction with 2U.

- **Information Technology Infrastructure for Research**. Over the past two decades I've played a leadership role in the Vanderbilt University Information Technology (VUIT) planning and governance processes. In addition to my latest role as the Associate Provost for Research, I've also chaired the faculty committee on Academic Computing and Information Technology (ACIT), served as the VUSE point of contact for VUIT, the VUSE representative for the Research IT committee as a member of the VUIT faculty advisory committee, as well as served as a member of the Research Advisory Committee on Information Technology (RACIT), and a member of the Provost's Research IT Special Project Working Group, which focuses on supporting the research needs of all schools at Vanderbilt.

# Appendix B

ALTHOS SIMPLIFYING KNOWLEDGE

VOIP Industry Dictionary is a free tool to get technical definitions and lookup industry terms from more than 11,000 terms, acronyms, and definitions.

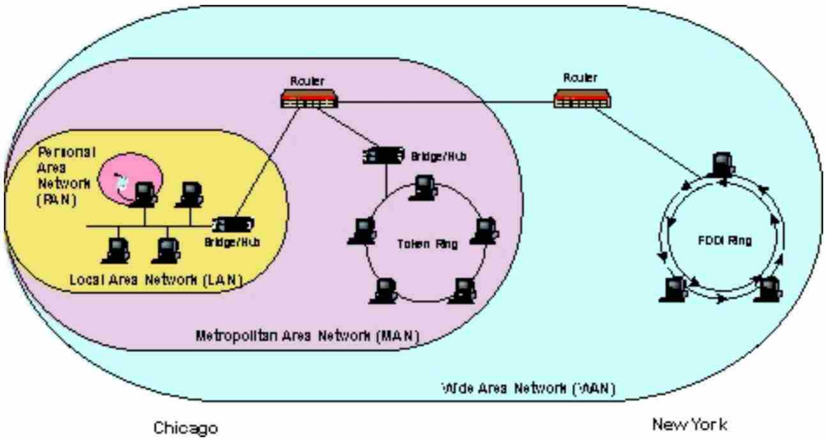




## Data Network

A data network is a system that **transfers data** between network access points (nodes) t**hrough data switching**, system control and interconnection transmission lines. Data networks are primarily designed to transfer data from one point to one or more points (multipoint). Data networks may be composed of a variety of communication systems including circuit switches, leased lines and packet switching networks. There are predominately two types of data networks, broadcast and point-to-point.

This figure shows the basic **types of data networks**. This diagram shows that the types of data networks range from very short range personal area networks (PANs) to wide area data networks (WANs). This example shows that a user is transferring data between a PDA and a computer within their **personal area network (PAN)** at their office in Chicago. This data can be transferred through their company's **local area network (LAN)**, through a city wide **metropolitan area network (MAN)** and through a **wide area network (WAN)** so it can reach other locations such as New York. When the data reaches New York, it is transferred back to a fiber distributed data interface (FDDI) local area network (LAN) so it can reach its destination computer.




Data Network Type Diagram

Lookup

This industry dictionary is powered by Althos Telecom Dictionary

## Add this Online Telecom Dictionary tool to your web site

In order to add this online TeleCom industry dictionary tool to your web site and give your visitors the way to lookup industry terms from more than 11,000 terms, acronyms, and definitions, just copy the HTML code below and put it into your web page where you want to provide access to your dictionary.

```
<table align="center" cellspacing="1" cellpadding="5" border="0"
bgcolor="#C0C0C0">
<tr bgcolor="#FFFFFF"><td><font style="font-size: 11px; font-
family: Verdana;"><b>Find TeleCom industry <br>terms,
acronyms, and definitions instantly:</b></font></td></tr>
<FORM METHOD="POST"
```

**Other Dictionaries:** Wireless Dictionary | Optical Dictionary | Billing Dictionary | Telecom Dictionary | CATV Dictionary | eMarketing Dictionary | IPTV Dictionary | VOIP Dictionary

**Related Websites:** Althos - Althos Books - Telecom Books - Telecom Dictionary

# Appendix C

Michael Duck and Richard Read

second edition

# Data Communications and Computer Networks

## for Computer Scientists and Engineers

PEARSON
Prentice Hall

# Data Communications and Computer Networks
## for Computer Scientists and Engineers

Second Edition

**Michael Duck**
and
**Richard Read**

PEARSON
Prentice Hall

Harlow, England • London • New York • Boston • San Francisco • Toronto • Sydney • Singapore • Hong Kong
Tokyo • Seoul • Taipei • New Delhi • Cape Town • Madrid • Mexico City • Amsterdam • Munich • Paris • Milan

 Wendt

**Pearson Education Limited**
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

*Visit us on the World Wide Web at:*
www.pearsoneduc.com

First published 1996
**Second edition 2003**



ISBN 0 130 93047 4

**British Library Cataloguing-in-Publication Data**
A catalogue record for this book is available from the British Library

**Library of Congress Cataloging-in-Publication Data**
Copy to follow

10 9 8 7 6 5 4 3 2 1
07 06 05 04 03

Typeset in 10/12pt Times by 35
Printed and bound in Great Britain by Henry Ling Ltd, at the Dorset Press, Dorchester, Dorset

Chapter 9

# Introduction to local area networks



LANs generally encompass a small physical area, no more than a few kilometres in diameter, and are usually confined within a single site. There may be as many as several hundred stations. LANs use relatively high signalling rates, originally of the order of 10 Mbps, but now typically 100 Mbps, and even as high as 1 Gbps. This chapter is an introduction to LANs and reviews the mechanisms to control station access to the network and also makes some comparisons of their performance. The next chapter will examine the standards for lower speed LAN operation up to a few tens of Mbps. Higher speed LANs, operating at 100 Mbps and beyond, are discussed in Chapter 11.

Messages within a LAN are transmitted as a series of variable length frames using transmission media which introduce only relatively low error rates. Since the length of the medium in a LAN is, unlike in WANs, relatively short, propagation delay tends to be short. Athough real-time services such as voice and video are carried over LANs, their frame-based transmissions tend to be more suited to data applications. This is because delay may be unpredictable and hence, on occasion, excessive for real-time use. In addition, transmission capacity has tended to be insufficient to meet the high-data-rate demands of video. However, with the emergence of 100 Mbps operation and beyond, coupled with advances in compression, real-time and multimedia operation is now feasible in many LANs.

A LAN comprises three hardware elements, namely a transmission medium, a controlling mechanism or protocol to govern access to the transmission medium, and an interface between the station and transmission medium. A software element is required to implement the medium access protocol for intercommunication between stations and for preparation of frames for transmission, and vice versa. These hardware and software elements, all of which are implemented at the lowest two layers of the

OSI Reference Model, are usually combined into a **Network Interface Card** (NIC), also known as a **Network Adapter**. An NIC interfaces with a standard computer such as a PC and at the physical layer connects to the particular transmission media to be used. Another software element is therefore also required to regulate the interface between the computer system and NIC.

## 9.1 Medium Access Control

The media used in LANs generally convey frames from only one station at a time, although the media themselves are generally shared by a number of stations. In order to overcome the difficulties which may arise through sharing, a **Medium Access Control** (MAC) mechanism or protocol is necessary. A MAC protocol merely regulates how stations may access the medium in an orderly fashion for correct operation and also attempts to ensure that each station obtains a fair share of its use.

LAN networks usually have only a single medium over which all messages, represented over a series of frames, are transmitted. If the medium is not being used two, or more, stations may simultaneously attempt an access, leading to a collision. An MAC technique is therefore required to regulate access by stations to the medium and handle the effect of two, or more, stations simlultaneously attempting to acces the medium. There is also the danger that once a pair of stations have established communication, all other stations may be excluded, perhaps indefinitely, or at least for a considerable period of time.

A LAN does not usually have any separate network control function for operation. Nor is a separate control function required to detect abnormal network conditions, or to control recovery therefrom. Rather, each station is generally equally responsible in a LAN, in which case control is said to be **fully distributed**.

Three general MAC techniques exist for use within fully distributed networks:

1. Contention: Here there is no regulating mechanism directly to govern stations attempting to access a medium. Rather, two or more stations may contend for the medium and any multiple simultaneous accesses are resolved as they arise.
2. Token passing: A single **token** exists within the network and is passed between stations in turn. Only a station holding the token may use the medium for transmission. This eliminates multiple simultaneous accesses of the medium with the attendant risk of collision.
3. Slotted and register insertion rings: Similar in principle to token passing, but a unique time interval is granted to a station for transmission.

### Carrier Sense Multiple Access

Many modern LANs evolved from a LAN known as **Aloha** which was one of the first primitive LANs to be developed. Aloha was packet based and used radio as its transmission medium. It was used by the University of Hawaii in the early 1970s to inter-

**JTAM** Job transfer, access and management.

**Link management** A function of the data link layer of the OSI Reference Model which is concerned with setting up and disconnection of a link.

**Local area network (LAN)** A data communications network used to interconnect a community of digital devices distributed over a localized area of up to, say, 10 $km^2$. The devices may be office workstations, mini- and microcomputers, intelligent instrumentation equipment, and so on.

**Logical Link Control (LLC)** A protocol forming part of the data link layer in LANs. It is concerned with the reliable transfer of data across the data link between two communicating systems.

**Management information base (MIB)** The name of the database used to hold the management information relating to a network or internetwork.

**Manchester encoding** A 1B2B code which converts each single binary 1 and binary 0 into two respective equal, and opposite, binary signal elements.

**Mark** A term traditionally used in telegraph systems to indicate a logic 1 state of a bit.

**Maximum burst size (MBS)** A traffic parameter, agreed between an ATM network and a customer, that is the agreed maximum size of a burst of cells with variable bit rate that can be accepted by the network.

**Media gateway** A device that converts media provided in one type of network to the format required for another type of network.

**Media Gateway Control Protocol** A protocol for controlling media gateways.

**Medium access control (MAC)** A method of determining which device has access to a shared transmission medium in a local area network.

**Metropolitan area network (MAN)** A network that links a set of LANs that are physically distributed around a town or city.

**Modem** The device that converts a binary (digital) data stream into an analogue (continuously varying) form, prior to transmission of the data across an analogue network (MODulator), and reconverts the received signal back into its binary form (DEModulator). Since each access port to the network normally requires a full-duplex (two-way simultaneous) capability, the device must perform both the MODulation and the DEModulation functions; hence the single name MODEM is used. As an example, a modem is normally required to transmit data across a telephone network.

**Moving Picture Experts Group (MPEG)** An ISO committee that generates standards for digital video compression. It also gives its name to their standards.

**Multidrop** A type of network configuration that supports more than two stations on the same transmission medium.

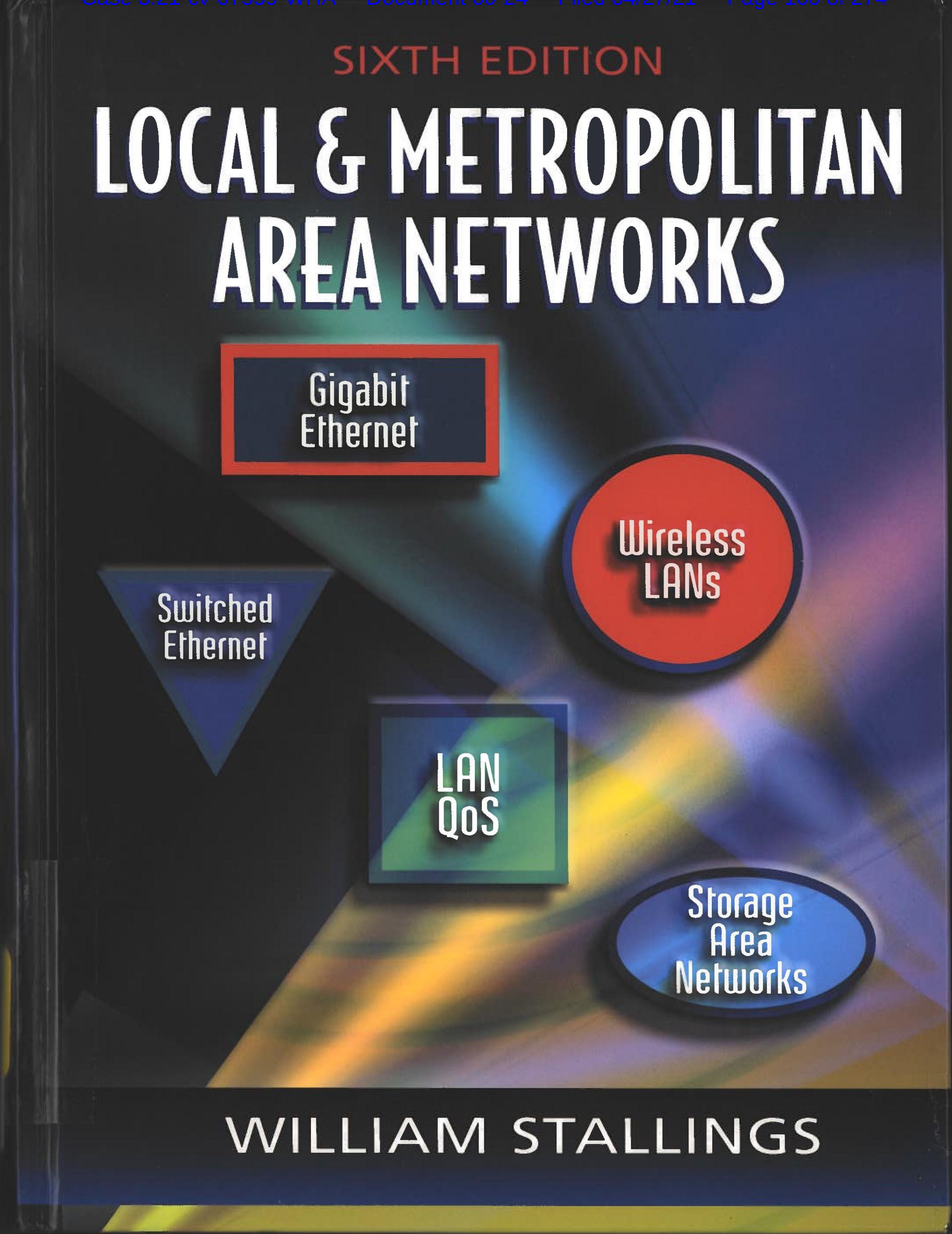
**Multiplexer** A device to enable a number of lower bit rate devices, normally situated in the same location, to share a single higher bit rate transmission line. The data-carrying capacity of the latter must be in excess of the combined bit rates of the low bit rate devices.

**Multi Protocol Label Switching** A standard that integrates layer 3 routing using IP addresses with a layer 2 switching technique such as ATM or Frame Relay.

**Network interface card** A physical interface in an end system such as a computer that connects to a transmission medium.

**Network layer** This corresponds to layer 3 of the ISO reference model for open systems interconnection. It is concerned with the establishment and clearing of logical or physical connections across the network being used.

# Appendix D


SIXTH EDITION
LOCAL & METROPOLITAN AREA NETWORKS
Gigabit Ethernet
Wireless LANs
Switched Ethernet
LAN QoS
Storage Area Networks
WILLIAM STALLINGS

# LOCAL AND METROPOLITAN AREA NETWORKS

SIXTH EDITION

William Stallings

Prentice Hall
Upper Saddle River, New Jersey 07458

**Library of Congress Cataloging-in-Publication Data**

Stallings, William.
Local and Metropolitan Area Networks / William Stallings.
p. cm.
Includes bibliographical references.
ISBN 0-13-012939-0
1. Computer networks. 2. LAN/MAN. I. Title.

TK5105.59.S725 1999
005.8—dc21 067829
CIP

*Editor-in-chief*: Marcia Horton
*Publisher*: Alan Apt
*Project manager*: Ana Terry
*Executive managing editor*: Vince O'Brien
*Managing editor*: David A. George
*Vice president of production and manufacturing*: David W. Riccardi
*Editorial assistant*: Toni Holm
*Editorial/production supervision*: Rose Kernan
*Art director*: Heather Scott
*Manufacturing buyer*: Pat Brown
*Marketing manager*: Jennie Burger



Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-012939-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada, Inc., *Toronto*
Prentice-Hall Hispanoamericana, S. A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Pearson Education Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

# CHAPTER 1

# INTRODUCTION

The local area network (LAN) has come to play a central role in information distribution and office functioning within businesses and other organizations. The major factors driving the widespread use of LANs have been the proliferation of personal computers, workstations, and servers, coupled with the increasing reliance on the client/server computing model.

With the dropping price of LAN hardware and software, LANs have become more numerous and larger, and they have taken on more and more functions within the organization. The upshot is that the LAN, once installed, quickly becomes almost as essential as the telephone system. At the same time, there is a proliferation of LAN types and options and a need to interconnect a number of LANs at the same site and with LANs at other sites. This has led to the development of LANs of higher and higher data rates and the relatively recent introduction of the metropolitan area network (MAN).

Before defining the terms LAN and MAN, it is useful to look at the trends responsible for the importance of these networks, which we do in the first section. Next we contrast the differences among LANs, MANs, and wide area networks (WANs). This is followed by a discussion of key application areas for LANs and MANs. This chapter also provides pointers to Internet resources relating to LANs and MANs.

## 1.1 THE NEED FOR LOCAL NETWORKS

Perhaps the driving force behind the widespread use of LANs and MANs is the dramatic and continuing decrease in computer hardware costs, accompanied by an increase in computer hardware capability. Year by year, the cost of computer systems continues to drop dramatically while the performance and capacity of those systems continue to rise equally dramatically. At a local warehouse club, you can pick up a personal computer for less than $1000 that packs the wallop of an IBM mainframe from 10 years ago. Inside that personal computer, including the microprocessor and memory and other chips, you get over 100 million transistors. You can't buy 100 million of anything else for so little. That many sheets of toilet paper would run more than $100,000.

Thus we have virtually "free" computer power. And this ongoing technological revolution has enabled the development of applications of astounding complexity and power. For example, desktop applications that require the great power of today's microprocessor-based systems include

- Image processing
- Speech recognition
- Videoconferencing
- Multimedia authoring
- Voice and video annotation of files

Workstation systems now support highly sophisticated engineering and scientific applications, as well as simulation systems, and the ability to apply workgroup

principles to image and video applications. In addition, businesses are relying on increasingly powerful servers to handle transaction and database processing and to support massive client/server networks that have replaced the huge mainframe computer centers of yesteryear.

All of these factors lead to an increased number of systems, with increased power, at a single site: office building, factory, operations center, and so on. At the same time, there is an absolute requirement to interconnect these systems to

- Share and exchange data among systems
- Share expensive resources

The need to share data is a compelling reason for interconnection. Individual users of computer resources do not work in isolation. They need facilities to exchange messages with other users, to access data from several sources in the preparation of a document or for an analysis, and to share project-related information with other members of a workgroup.

The need to share expensive resources is another driving factor in the development of networks. The cost of processor hardware has dropped far more rapidly than the cost of mass storage devices, video equipment, printers, and other peripheral devices. The result is a need to share these expensive devices among a number of users to justify the cost of the equipment. This sharing requires some sort of client/server architecture operating over a network that interconnects users and resources.

## 1.2 LANs, MANs, AND WANs

LANs, MANs, and WANs are all examples of communications networks. A communications network is a facility that interconnects a number of devices and provides a means for transmitting data from one attached device to another.

There are a number of ways of classifying communications networks. One way is in terms of the technology used: specifically, in terms of topology and transmission medium. That approach is explored in Chapter 4. Perhaps the most commonly used means of classification is on the basis of geographical scope. Traditionally, networks have been classified as either LANs or WANs. A term that is sometimes used is the MAN.

Figure 1.1 illustrates these categories, plus some special cases. By way of contrast, the typical range of parameters for a multiple-processor computer is also depicted.

### Wide Area Network

WANs have traditionally been considered to be those that cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device.

Traditionally, WANs have provided only relatively modest capacity to subscribers. For data attachment, either to a data network or to a telephone network

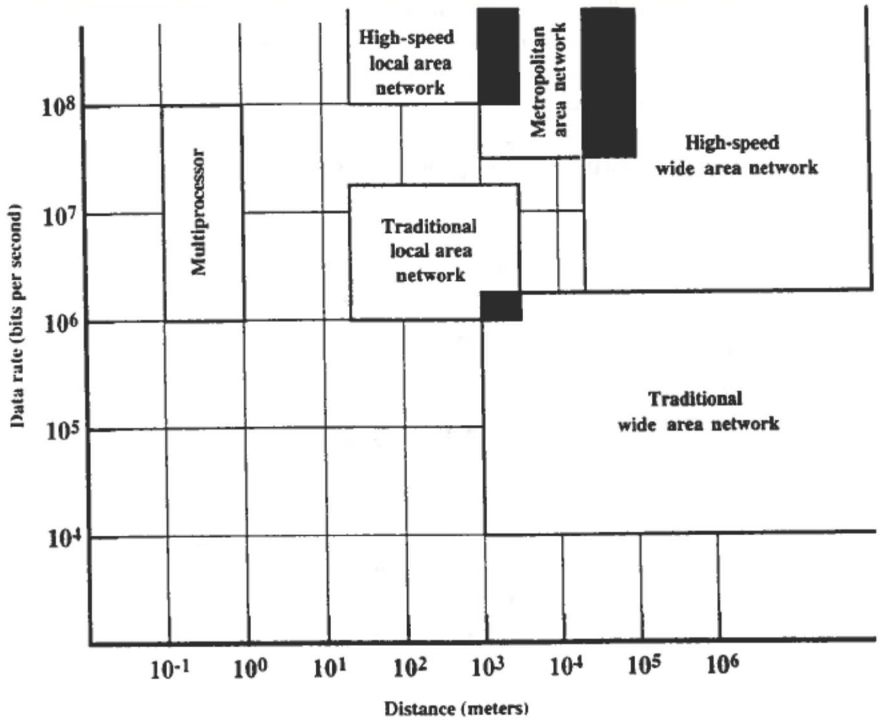



**Figure 1.1** Comparison of Multiprocessor Systems, LANs, MANs, and WANs

by means of a modem, data rates of 9600 bps or even less have been common. Business subscribers have been able to obtain higher rates, with a service known as T1, which operates at 1.544 Mbps, being common. The most important recent development in WANs in this range of performance has been the development of the integrated services digital network (ISDN), which provides circuit-switching and packet-switching services at rates up to 1.544 Mbps (2.048 Mbps in Europe).

The continuing development of practical optical fiber facilities has led to the standardization of much higher data rates for WANs, and these services are becoming more widely available. These high-speed WANs provide user connections in the 10s and 100s of Mbps, using transmission techniques known as frame relay and asynchronous transfer mode (ATM).

## Local Area Network

As with WANs, a LAN is a communications network that interconnects a variety of devices and provides a means for information exchange among those devices. There are several key distinctions between LANs and WANs:

1. The scope of the LAN is small, typically a single building or a cluster of buildings. This difference in geographic scope leads to different technical solutions, as we shall see.

2. It is usually the case that the LAN is owned by the same organization that owns the attached devices. For WANs, this is less often the case, or at least a significant fraction of the network assets are not owned. This has two implications. First, care must be taken in the choice of LAN, since there may be a substantial capital investment (compared to dial-up or leased charges for WANs) for both purchase and maintenance. Second, the network management responsibility for a LAN falls solely on the user.
3. The internal data rates of LANs are typically much greater than those of WANs.

LANs have been the focus of a standardization effort by the IEEE 802 committee, and it is useful to review the IEEE definition of a LAN (Table 1.1).

### A Simple LAN

A simple example of a LAN that highlights some of its characteristics is shown in Figure 1.2. All of the devices are attached to a shared transmission medium. A transmission from any one device can be received by all other devices attached to the same network.

What is not apparent in Figure 1.2 is that each device attaches to the LAN through a hardware/software module that handles the transmission and medium access functions associated with the LAN. Typically, this module is implemented as a physically distinct network interface card (NIC) in each attached device. The NIC contains the logic for accessing the LAN and for sending and receiving blocks of data on the LAN.

An important function of the NIC is that it uses a buffered transmission technique to accommodate the difference in the data rate between the LAN medium

**Table 1.1** Definitions of LANs and MANs*

| |
|---|
| The LANs described herein are distinguished from other types of data networks in that they are optimized for a moderate size geographic area such as a single office building, a warehouse, or a campus. The IEEE 802 LAN is a shared medium peer-to-peer communications network that broadcasts information for all stations to receive. As a consequence, it does not inherently provide privacy. The LAN enables stations to communicate directly using a common physical medium on a point-to-point basis without any intermediate switching node being required. There is always need for an access sublayer in order to arbitrate the access to the shared medium. The network is generally owned, used, and operated by a single organization. This is in contrast to Wide Area Networks (WANs) that interconnect communication facilities in different parts of a country or are used as a public utility. These LANs are also different from networks, such as backplane buses, that are optimized for the interconnection of devices on a desk top or components within a single piece of equipment. |
| A MAN is optimized for a larger geographical area than a LAN, ranging from several blocks of buildings to entire cities. As with local networks, MANs can also depend on communications channels of moderate-to-high data rates. Error rates and delay may be slightly higher than might be obtained on a LAN. A MAN might be owned and operated by a single organization, but usually will be used by many individuals and organizations. MANs might also be owned and operated as public utilities. They will often provide means for internetworking of local networks. Although not a requirement for all LANs, the capability to perform local networking of integrated voice and data (IVD) devices is considered an optional function for a LAN. Likewise, such capabilities in a network covering a metropolitan area are optional functions of a MAN. |

* From IEEE 802 Standard, *Local and Metropolitan Area Networks: Overview and Architecture*, 1990.

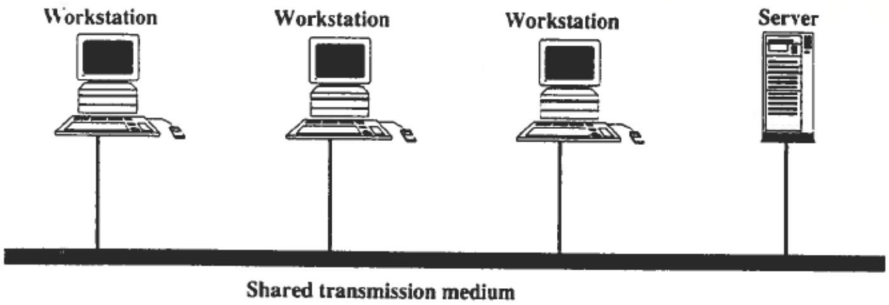



**Figure 1.2** Simple Local Area Network

and the NIC-processor link, as illustrated in Figure 1.3. The NIC captures transmissions intended for the attached device, which arrive at the data rate of the LAN, which may be, for example, 10 Mbps. When a block of data is captured, it is stored temporarily in an input buffer. It is then delivered to the host processor, often over some sort of backplane bus, at the data rate of that bus. This data rate is typically different from the LAN data rate. For example, it may be 50 or 100 Mbps. Thus the NIC acts as an adapter between the data rate on the host system bus and the data rate on the LAN.

### High-Speed LANs

Traditional LANs have provided data rates in a range from about 1 to 20 Mbps. These data rates, though substantial, have become increasingly inadequate

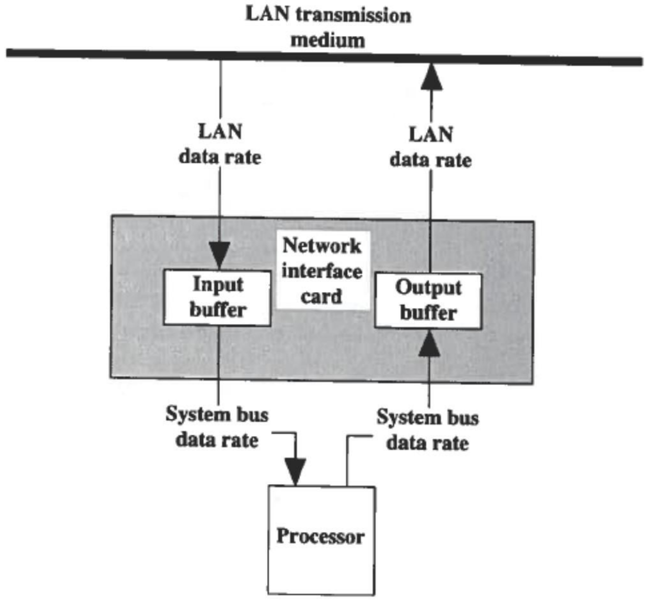



**Figure 1.3** Buffered Transmission through a Network Interface Card

with the proliferation of devices, the growth in multimedia applications, and the increased use of the client/server architecture. As a result, much of the effort in LAN development has been in the development of high-speed LANs, with data rates of 100 Mbps or more. In later chapters, we will see a number of examples of high-speed LANs.

### Metropolitan Area Networks

As the name suggests, a MAN occupies a middle ground between LANs and WANs. Interest in MANs has come about as a result of a recognition that the traditional point-to-point and switched network techniques used in WANs may be inadequate for the growing needs of organizations. While frame relay and ATM promise to meet a wide range of high-speed needs, there is a requirement now for both private and public networks that provide high capacity at low costs over a large area. The high-speed shared-medium approach of the LAN standards provides a number of benefits that can be realized on a metropolitan scale. As Figure 1.1 indicates, MANs cover greater distances at higher data rates than LANs, although there is some overlap in geographical coverage.

The primary market for MANs is the customer that has high-capacity needs in a metropolitan area. A MAN is intended to provide the required capacity at lower cost and greater efficiency than obtaining an equivalent service from the local telephone company.

## 1.3 APPLICATIONS OF LANS AND MANS

The variety of applications for LANs and MANs is wide. To provide some insight into the types of requirements that LANs and MANs are intended to meet, this section provides a brief discussion of some of the most important general application areas for these networks.

### Personal Computer Local Networks

A common LAN configuration is one that supports personal computers. With the relatively low cost of such systems, individual managers within organizations often independently procure personal computers for departmental applications, such as spreadsheet and project management tools, and Internet access.

But a collection of department-level processors will not meet all of an organization's needs; central processing facilities are still required. Some programs, such as econometric forecasting models, are too big to run on a small computer. Corporate-wide data files, such as accounting and payroll, require a centralized facility but should be accessible to a number of users. In addition, there are other kinds of files that, although specialized, must be shared by a number of users. Further, there are sound reasons for connecting individual intelligent workstations not only to a central facility but to each other as well. Members of a project or organization team need to share work and information. By far the most efficient way to do so is digitally.

Certain expensive resources, such as a disk or a laser printer, can be shared by all users of the departmental LAN. In addition, the network can tie into larger corporate network facilities. For example, the corporation may have a building-wide LAN and a wide area private network. A communications server can provide controlled access to these resources.

LANs for the support of personal computers and workstations have become nearly universal in organizations of all sizes. Even those sites that still depend heavily on the mainframe have transferred much of the processing load to networks of personal computers. Perhaps the prime example of the way in which personal computers are being used is to implement client/server applications.

For personal computer networks, a key requirement is low cost. In particular, the cost of attachment to the network must be significantly less than the cost of the attached device. Thus, for the ordinary personal computer, an attachment cost in the hundreds of dollars is desirable. For more expensive, high-performance workstations, higher attachment costs can be tolerated. In any case, this suggests that the data rate of the network may be limited; in general, the higher the data rate, the higher the cost.

### Backend Networks and Storage Area Networks

Backend networks are used to interconnect large systems such as mainframes, supercomputers, and mass storage devices. The key requirement here is for bulk data transfer among a limited number of devices in a small area. High reliability is generally also a requirement. Typical characteristics include the following:

- **High data rate:** To satisfy the high-volume demand, data rates of 100 Mbps or more are required.
- **High-speed interface:** Data transfer operations between a large host system and a mass storage device are typically performed through high-speed parallel I/O interfaces, rather than slower communications interfaces. Thus, the physical link between station and network must be high speed.
- **Distributed access:** Some sort of distributed medium access control (MAC) technique is needed to enable a number of devices to share the medium with efficient and reliable access.
- **Limited distance:** Typically, a backend network will be employed in a computer room or a small number of contiguous rooms.
- **Limited number of devices:** The number of expensive mainframes and mass storage devices found in the computer room generally numbers in the tens of devices.

. Typically, backend networks are found at sites of large companies or research installations with large data processing budgets. Because of the scale involved, a small difference in productivity can mean millions of dollars.

Consider a site that uses a dedicated mainframe computer. This implies a fairly large application or set of applications. As the load at the site grows, the existing mainframe may be replaced by a more powerful one, perhaps a multiprocessor system. At some sites, a single-system replacement will not be able to keep up;

# PART TWO LAN/MAN Architecture

Part Two surveys the key technology elements that are common to all types of LANs and MANs, including topology, transmission medium, medium access control, and logical link control.

## CHAPTER 4 TOPOLOGIES AND TRANSMISSION MEDIA

The essential technology underlying all forms of LANs and MANs comprises topology, transmission medium, and medium access control technique. Chapter 4 examines the first two of these elements. Four topologies are in common use: bus, tree, ring, and star. The most common transmission media for local networking are twisted pair (unshielded and shielded), coaxial cable (baseband and broadband), optical fiber, and wireless. The chapter closes with a discussion of structured cabling systems.

## CHAPTER 5 PROTOCOL ARCHITECTURE

Chapter 5 introduces the protocols needed for stations attached to a LAN to cooperate with each other in the exchange of data. Specifically, the chapter provides an overview of link control and medium access control protocols. The use of bridges and routers to interconnect LANs is also introduced.

# CHAPTER 6 LOGICAL LINK CONTROL

Logical link control (LLC) is the highest layer that is specifically part of the LAN/MAN protocol architecture. It is used above all of the medium access control (MAC) standards. The primary purpose of this layer is to provide a means of exchanging data between end users across a link or a collection of LANs interconnected by bridges. Different forms of the LLC service are specified to meet specific reliability and efficiency needs. After a discussion of these services, Chapter 6 deals with some of the key mechanisms of link control protocols. Finally, the specific LLC protocols are examined.

# Appendix E

Comer

CD-ROM by Droms

# Computer Networks and Internets

Second Edition

# Computer Networks And Internets

## Second Editon

Douglas E. Comer
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

PRENTICE HALL
Upper Saddle River, New Jersey 07458

**Library of Congress Cataloging-in-Publications Data**

Comer, Douglas.
Computer networks and internets / Douglas E. Comer. -- 2nd ed.
p. cm.
Includes index.
ISBN 0-13-083617-6
1. Computer networks. 2. Internetworking (Telecommunication)
I. Title.
TK5105.5.C5897 1999
004.6--dc21 98-47110
CIP

Publisher: **ALAN APT**
Development Editor: **SONDRA CHAVEZ**
Editor-in-chief: **MARCIA HORTON**
Production editor: **MIMI JETT—Interactive Composition Corporation**
Managing editor: **EILEEN CLARK**
Director of production and manufacturing: **DAVID W. RICCARDI**
Cover director: **HEATHER SCOTT**
Manufacturing buyer: **DONNA M. SULLIVAN**
Manufacturing manager: **TRUDY PISCIOTTI**
Editorial assistant: **TONI HOLM**



The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with or arising out of the furnishing, performance, or use of these programs.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-083617-6

Prentice-Hall International (UK) Limited, London
Prentice-Hall of Australia Pty. Limited, Sydney
Prentice-Hall Canada Inc., Toronto
Prentice-Hall Hispanoamericana, S.A., Mexico
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Simon & Schuster Asia Pte. Ltd., Singapore
Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

## 7.3 Shared Communication Channels

The history of computer networking changed dramatically during the late 1960s and early 1970s when researchers developed a form of computer communication known as *Local Area Networks* (*LAN*s). Devised as alternatives to expensive, dedicated point-to-point connections, the designs differ fundamentally from long-distance networks because they rely on sharing the network. Each LAN consists of a single shared medium, usually a cable, to which many computers attach. The computers take turns using the medium to send packets.

Several LAN designs emerged from the research. The designs differ in details such as the voltages and modulation techniques used, and the approach to sharing (i.e., the mechanisms used to coordinate access and transmit packets).

Because it eliminates duplication, sharing has an important economic impact on networking: it reduces cost. Consequently, Local Area Network technologies that allow a set of computers to share a medium have become popular. In fact,

> *Networks that allow multiple computers to share a communication medium are used for local communication. Point-to-point connections are used for long-distance networks and a few other special cases.*

If sharing reduces cost, why are shared networks used only for local communication? Both technical and economic reasons contribute to the answer. We said that the computers attached to a shared network must coordinate use of the network. Because coordination requires communication and the time required to communicate depends on distance, a large geographic separation between computers introduces longer delays. Thus, shared networks with long delays are inefficient because they spend more time coordinating use of the shared medium and less time sending data. In addition, engineers have learned that providing a high bandwidth communication channel over long distances is significantly more expensive than providing the same bandwidth communication over a short distance.

## 7.4 Significance Of LANs And Locality Of Reference

The significance of LANs can be stated simply:

> *Local Area Network technologies have become the most popular form of computer networks. LANs now connect more computers than any other type of network.*

One of the reasons so many LANs have been installed is economic: LAN technologies are both inexpensive and widely available. However, the main reason the demand for LANs is high can be attributed to a fundamental principle of computer networking known as *locality of reference*. The locality of reference principle states that communi-

cation among a set of computers is not random, but instead follows two patterns. First, if a pair of computers communicates once, the pair is likely to communicate again in the near future and then periodically. The pattern is called *temporal locality of reference* to imply a relationship over time. Second, a computer tends to communicate most often with other computers that are nearby. The second pattern is called *physical locality of reference*† to emphasize the geographic relationship. We can summarize:

> The locality of reference principle: *computer communication follows two distinct patterns. First, a computer is more likely to communicate with computers that are physically nearby than with computers that are far away. Second, a computer is more likely to communicate with the same set of computers repeatedly.*

The locality of reference principle is easy to understand because it applies to human communication. For example, people communicate most often with others who are physically nearby (e.g., working together). Furthermore, if an individual communicates with someone (e.g., a friend or family member), the individual is likely to communicate with the same person again.

## 7.5 LAN Topologies

Because many LAN technologies have been invented, it is important to know how specific technologies are similar and how they differ. To help understand similarities, each network is classified into a category according to its *topology* or general shape. This section describes the three topologies used most often with LANs; later sections add more detail and show specific examples.

### 7.5.1 Star Topology

A network uses a *star topology* if all computers attach to a central point. Figure 7.3 illustrates the concept.

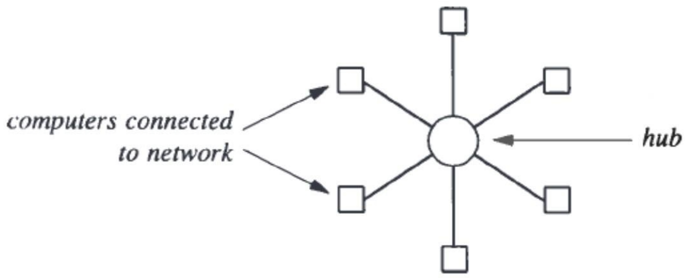



**Figure 7.3** Illustration of the star topology in which each computer attaches to a central point called a *hub*.

†Physical locality of reference is sometimes referred to as *spatial locality of reference*.

Because a star-shaped network resembles the spokes of a wheel, the center of a star network is often called a *hub*. A typical hub consists of an electronic device that accepts data from a sending computer and delivers it to the appropriate destination.

Figure 7.3 illustrates an idealized star network. In practice, star networks seldom have a symmetric shape in which the hub is located an equal distance from all computers. Instead, a hub often resides in a location separate from the computers attached to it. For example, Chapter 9 will illustrate that computers can reside in individual offices, while the hub resides in a location accessible to an organization's networking staff.

### 7.5.2 Ring Topology

A network that uses a *ring topology* arranges for computers to be connected in a closed loop – a cable connects the first computer to a second computer, another cable connects the second computer to a third, and so on, until a cable connects the final computer back to the first. The name *ring* arises because one can imagine the computers and the cables connecting them arranged in a circle as Figure 7.4 illustrates.

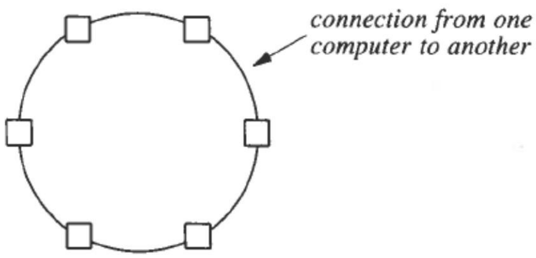



**Figure 7.4** Illustration of a ring topology in which computers are connected in a closed loop. Each computer connects directly to two others.

It is important to understand that the *ring*, like the star topology, refers to logical connections among computers, not physical orientation – the computers and connections in a ring network need not be arranged in a circle. Instead, the cable between a pair of computers in a ring network may follow a hallway or rise vertically from one floor of a building to another. Furthermore, if one computer is far from others in the ring, the two cables that connect the distant computer may follow the same physical path.

### 7.5.3 Bus Topology

A network that uses a *bus topology* usually consists of a single, long cable to which computers attach†. Any computer attached to a bus can send a signal down the cable, and all computers receive the signal. Figure 7.5 illustrates the topology. Because all computers attached to the cable can sense an electrical signal, any computer can send

†In practice, the ends of a bus network must be terminated to prevent electrical signals from reflecting back along the bus.

data to any other computer. Of course, the computers attached to a bus network must coordinate to ensure that only one computer sends a signal at any time or chaos results.

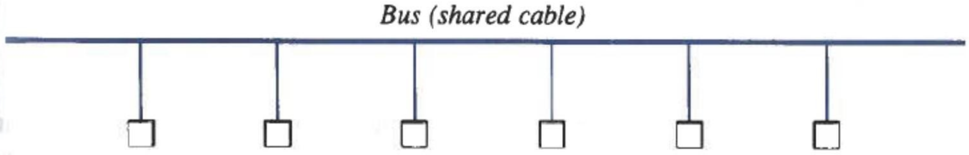



**Figure 7.5** Illustration of a bus topology in which all computers attach to a single cable.

### 7.5.4 The Reason For Multiple Topologies

Each topology has advantages and disadvantages. A ring topology makes it easy for computers to coordinate access and to detect whether the network is operating correctly. However, an entire ring network is disabled if one of the cables is cut. A star topology helps protect the network from damage to a single cable because each cable connects only one machine. A bus requires fewer wires than a star, but has the same disadvantage as a ring: a network is disabled if someone accidentally cuts the main cable. In addition to later sections in this chapter, other chapters provide detailed examples of network technologies that illustrate some of the differences.

We can summarize the major points about network topologies.

> *Networks are classified into broad categories according to their general shape. The primary topologies used with LANs are star, ring, and bus; each topology has advantages and disadvantages.*

## 7.6 Example Bus Network: Ethernet

### 7.6.1 History Of The Ethernet

*Ethernet* is a well-known and widely used network technology that employs bus topology. Ethernet was invented at Xerox Corporation's Palo Alto Research Center in the early 1970s. Digital Equipment Corporation, Intel Corporation, and Xerox later cooperated to devise a production standard, which is informally called *DIX Ethernet* for the initials of the three companies. IEEE now controls Ethernet standards†. In its original version, an Ethernet LAN consisted of a single coaxial cable, called the *ether*, to which multiple computers connect. Engineers use the term *segment* to refer to the Ethernet coaxial cable. A given Ethernet segment is limited to *500* meters in length, and the standard requires a minimum separation of *3* meters between each pair of connections.

†Several variations of Ethernet currently exist; this section describes the original technology and leaves the discussion of alternatives until Chapter 9.

The original Ethernet hardware operated at a bandwidth of 10 Megabits per second (Mbps); a later version known as *Fast Ethernet* operates at 100 Mbps, and the most recent version, which is known as *Gigabit Ethernet* operates at 1000 Mbps or 1 Gigabit per second (Gbps).

### 7.6.2 Sharing On An Ethernet

The Ethernet standard specifies all details, including the format of frames that computers send across the ether†, the voltage to be used, and the method used to modulate a signal.

Because it uses a bus topology, Ethernet requires multiple computers to share access to a single medium. A sender transmits a signal, which propagates from the sender toward both ends of the cable. Figure 7.6 illustrates how data flows across an Ethernet.

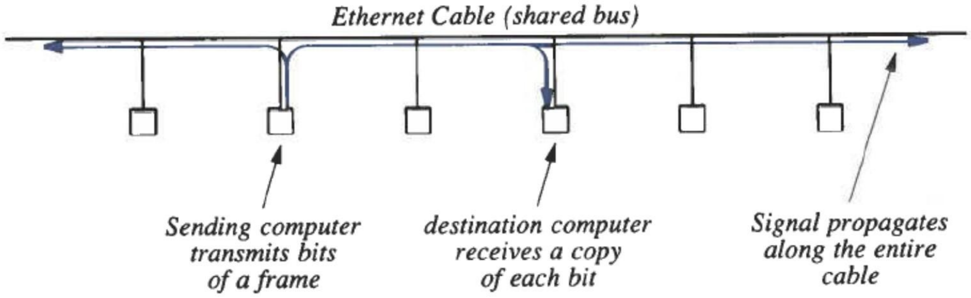



**Figure 7.6** Conceptual flow of bits across an Ethernet. While transmitting a frame, a computer has exclusive use of the cable.

As the figure shows, a signal propagates from the sending computer to both ends of the shared cable. It is important to understand that sharing in local area networks technologies, does not mean that multiple frames are being sent at the same time. Instead, the sending computer has exclusive use of the entire cable during the transmission of a given frame – other computers must wait. After one computer finishes transmitting one frame, the shared cable becomes available for another computer to use. To summarize:

> *Ethernet is a bus network in which multiple computers share a single transmission medium. While one computer transmits a frame to another, all other computers must wait.*

†Chapter 8 discusses Ethernet frames in more detail and shows an example.

## 7.7 Carrier Sense On Multi-Access Networks (CSMA)

The most interesting aspect of Ethernet is the mechanism used to coordinate transmission. An Ethernet network does not have a centralized controller that tells each computer how to take turns using the shared cable. Instead, all computers attached to an Ethernet participate in a distributed coordination scheme called *Carrier Sense Multiple Access* (*CSMA*). The scheme uses electrical activity on the cable to determine status. When no computer is sending a frame, the ether does not contain electrical signals. During frame transmission, however, a sender transmits electrical signals used to encode bits. Although the signals differ slightly from the carrier waves described in Chapter 5, they are informally called a *carrier*. Thus, to determine whether the cable is currently being used, a computer can check for a carrier. If no carrier is present, the computer can transmit a frame. If a carrier is present, the computer must wait for the sender to finish before proceeding. Technically, checking for a carrier wave is called *carrier sense*, and the idea of using the presence of a signal to determine when to transmit is called *Carrier Sense Multiple Access* (*CSMA*).

## 7.8 Collision Detection And Backoff With CSMA/CD

Because CSMA allows each computer to determine whether a shared cable is already in use by another computer, it prevents a computer from interrupting an ongoing transmission. However, CSMA cannot prevent all possible conflicts. To understand why, imagine what happens if two computers at opposite ends of an idle cable both have a frame ready to send at the same time. When they check for a carrier, both stations find the cable idle, and both start to send frames simultaneously. The signals travel at approximately *70%* of the speed of light, and when the signals transmitted by two computers reach the same point on the cable, they interfere with each other.

The interference between two signals is called a *collision*. Although a collision does not harm the hardware, it produces a garbled transmission that prevents either of the two frames from being received correctly. To ensure that no other computer transmits simultaneously, the Ethernet standard requires a sending station to monitor signals on the cable. If the signal on the cable differs from the signal that the station is sending, it means that a collision has occurred†. Whenever a collision is detected, a sending station immediately stops transmitting. Technically, monitoring a cable during transmission is known as *Collision Detect* (*CD*), and the Ethernet mechanism is known as *Carrier Sense Multiple Access with Collision Detect* (*CSMA/CD*).

CSMA/CD does more than merely detect collisions – it also recovers from them. After a collision occurs, a computer must wait for the cable to become idle again before transmitting a frame. However, if the computers begin to transmit as soon as the ether becomes idle, another collision will occur. To avoid multiple collisions, Ethernet requires each computer to delay after a collision before attempting to retransmit. The standard specifies a maximum delay, *d*, and forces each computer to choose a random delay less than *d*. In most cases, when a computer chooses a delay at random, it will

†To guarantee that a collision has time to reach all stations before they stop transmitting, the Ethernet standard specifies both a maximum cable length and a minimum frame size.

select a value that differs from any of the values chosen by the other computers – the computer that chooses the smallest delay will proceed to send a frame and the network will return to normal operation.

If two or more computers happen to choose nearly the same amount of delay after a collision, they will both begin to transmit at nearly the same time, producing a second collision. To avoid a sequence of collisions, Ethernet requires each computer to double the range from which a delay is chosen after each collision. Thus, a computer chooses a random delay from *0* to *d* after one collision, a random delay between *0* and *2d* after a second collision, between *0* and *4d* after a third, and so on. After a few collisions, the range from which a random value is chosen becomes large, and the probability is high that some computer will choose a short delay and transmit without a collision.

Technically, doubling the range of the random delay after each collision is known as *binary exponential backoff*. In essence, exponential backoff means that an Ethernet can recover quickly after a collision because each computer agrees to wait longer times between attempts when the cable becomes busy. In the unlikely event that two or more computers choose delays that are approximately equal, exponential backoff guarantees that contention for the cable will be reduced after a few collisions. We can summarize:

> *Computers attached to an Ethernet use CSMA/CD in which a computer waits for the ether to be idle before transmitting a frame. If two computers transmit simultaneously, a collision occurs; the computers use exponential backoff to choose which computer will proceed. Each computer delays a random time before trying to transmit again, and then doubles the delay for each successive collision.*

## 7.9 Wireless LANs And CSMA/CA

A set of *wireless LAN* technologies are available that use a modified form of CSMA/CD. The products, which are manufactured by several companies are available under a variety of trade names. For example, NCR Corporation sells *WaveLAN*, Solectek sells *AirLAN*, and Proxim Corporation sells *RangeLAN*.

Instead of transmitting signals across a cable, wireless LAN hardware uses antennas to broadcast RF signals through the air, which other computers receive. The devices use 900 MHz frequencies to permit data to be sent at 2 Mbps. Like other LAN technologies, the wireless LANs use sharing. That is, all the computers participating in a given wireless LAN are configured to the same radio frequency. Thus, they must take turns sending packets.

One difference between the way wired and wireless LANs manage sharing arises because of the way wireless transmissions propagate. Although the electromagnetic energy radiates in all directions, wireless LAN transmitters use low power, meaning that a transmission only has enough power to travel a short distance. Furthermore, metallic obstructions can block the signal. Thus, wireless units located far apart or behind obstructions will not receive each other's transmissions.

easily carry data in two directions simultaneously, each connection uses a pair of fibers as Figure 7.11 illustrates.

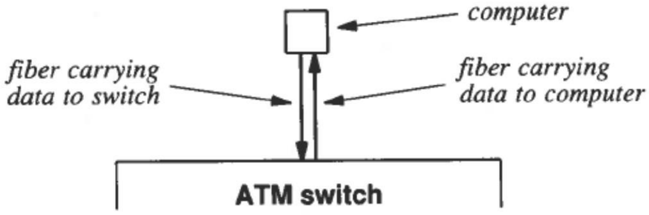



**Figure 7.11** Details of a connection between an ATM switch and a computer. Each connection consists of a pair of optical fibers. One fiber carries data to the switch, and the other carries data to the computer.

Like the optical fibers used with FDDI, the pair of fibers used to connect a computer to an ATM switch are fastened together. Usually, the jacket on one fiber contains a colored stripe or is labeled; whoever installs a connection uses the label to ensure that the output of the switch connects to the input of the computer and vice versa.

To summarize:

> *An ATM network is formed from a switch to which multiple computers attach. The connection between a computer and an ATM switch consists of a pair of fibers, one carrying data in each direction.*

## 7.14 Summary

This chapter discusses an alternative to direct point-to-point communication called a Local Area Network (LAN). Designed for use over a small distance (e.g., in a building), a LAN does not need a separate wire between each pair of computers. Instead, a LAN consists of a single, shared medium to which many computers attach. The computers take turns using the medium to send data.

Although LAN technologies require computers to divide data into small packets called frames, only one packet can be transmitted on a LAN at any time. That is, while transmitting, a computer has exclusive use of the LAN. To make access fair, each computer is permitted to hold the shared medium for the transmission of one frame before allowing another computer to proceed. Thus, after it gains control, a computer sends a frame and then relinquishes control to another computer.

Each computer network can be classified into one of a few basic categories, depending on its topology. A bus topology consists of a single, shared cable to which many computers attach. When it uses a bus, a computer transmits a signal that all other computers attached to the bus receive. A ring topology consists of computers connected in a closed loop. The first computer connects to the second, the second connects to the third, and so on, until the last computer connects back to the first. Finally, a star topology resembles a wheel with the network itself corresponding to a central hub, and the links to individual computers corresponding to spokes. Each topology has advantages and disadvantages; no topology is best for all purposes.

LAN technologies exist that use each topology. An Ethernet LAN uses a bus topology, as does LocalTalk. To access an Ethernet, stations obey Carrier Sense Multiple Access with Collision Detect (CSMA/CD). That is, a station waits for the ether to be idle, and then attempts to send. If two stations transmit at the same time, a collision results, causing them to wait a random time before trying again. Successive collisions cause exponential backoff in which each station doubles its delay.

Wireless LANs such as WaveLAN, RangeLAN, or AirLAN use Carrier Sense Multiple Access With Collision Avoidance (CSMA/CA). Before transmitting a data frame, a sender transmits a small control message to which the receiver responds. The exchange of control messages notifies all stations within range of the receiver that a data transmission is about to occur. Other stations then remain silent while the transmission takes place (i.e., avoid a collision), even if they do not receive a copy of the signal.

Stations attached to a token passing ring network also share the medium. While one station transmits a frame, all other stations pass the bits around the ring, which allows the sender to verify that the bits were transmitted correctly. To coordinate use of the ring and guarantee fairness, stations on a token ring send a special message called a token. A station waits for the token to arrive, uses the complete ring to transmit one frame, and then sends the token to the next station. IBM Token Ring and FDDI networks both use token passing. FDDI differs from conventional token passing technologies because it can be configured with an extra ring that is used to recover from catastrophic failures. The extra ring is called counter-rotating because data flows the opposite direction than on the main ring. An FDDI network with a counter-rotating ring is said to be self-healing because it can detect a failure and loop back along the reverse ring to close the path.

LANs that use ATM technology have a star topology. An ATM switch forms the hub of the star to which each computer connects. Because ATM is designed to operate at high speed, the connection between a computer and an ATM switch uses a pair of optical fibers, with one fiber carrying data in each direction.

# Appendix F

# WEBSTER'S NEW WORLD™

# TELECOM DICTIONARY

*A comprehensive reference for telecommunications terminology*

*More than 4,600 telecommunications terms and acronyms defined and explained*

*Clear, up-to-date coverage of recent data communications terminology, including VoIP, SANs, 802.11n, and Bluetooth*

RAY HORAK

WE DEFINE YOUR WORLD®

# WEBSTER'S NEW WORLD™

# Telecom Dictionary

Ray Horak

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

Wiley Publishing, Inc.

**Webster's New World® Telecom Dictionary**

Published by
**Wiley Publishing, Inc.**
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com



Published simultaneously in Canada

ISBN: 978-0-471-77457-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Horak, Ray.
Webster's New World telecom dictionary / Ray Horak.
p. cm.
ISBN 978-0-471-77457-0 (pbk.)
1. Telecommunication—Dictionaries. I. Title.
TK5102.H65 2007
621.38203—dc22
2007024232

dialing the destination telephone number. Synonymous with *caller ID*. See also *ACD*, *ANI*, *call center*, *CLASS*, *LEC*, *PSTN*, and *screen pop*. **2.** A voice telephone system feature that supports the CLID network service and offers a similar capability for station-to-station PBX calls. See also *CLASS*.

**client** In a client/server architecture, a complete, standalone computer that optimizes the user interface, relying on servers to handle the more mundane tasks associated with application and file storage, network administration, security, and other critical functions. See also *architecture*, *client/server*, and *server*.

**client mesh** See *pure mesh*.

**client/server** A network architecture that distributes intelligence and responsibilities at several levels, with some machines designated as servers to serve the needs of client machines. A server can be a mainframe, minicomputer, or personal computer that operates in a time-sharing mode to provide for the needs of many clients. Client machines are complete, standalone computers that optimize the user interface, relying on servers to handle the more mundane tasks associated with application and file storage, network administration, security, and other critical functions. See also *peer-to-peer*.

**Clipper Chip** An integrated circuit that uses the Skipjack voice encryption algorithm developed by the United States National Security Agency (NSA) for the National Institute of Science and Technology (NIST). Skipjack is a block coding algorithm that encrypts 64-bit data blocks with an 80-bit key. Data encrypted by the Skipjack algorithm can be provided not only to the intended recipient through the use of a key, but also by the U.S. government through the use of a back door into a Law Enforcement Access Field (LEAF). The Clipper Chip is manufactured by the U.S. government, which has tried unsuccessfully to make it, and similar technologies, mandatory for voice encryption in the United States. Privacy advocates feared that government authorities would abuse the back door. Law enforcement authorities fear that the widespread use of other voice encryption technologies will make it impossible to place legal wiretaps. See also *algorithm*, *back door*, *encryption*, *integrated circuit*, and *wiretap*.

**CLNP (ConnectionLess Network Protocol)** A Network Layer datagram protocol from the International Organization for Standardization (ISO) for use over OSI (Open Systems Integration) networks and specified in ISO 8473. CLNP is very similar to Internet Protocol (IP). The datagram size is the same as IP, and there are similar mechanisms for fragmentation, error control, and lifetime control. CLNP, however, has an address space of 20 octets compared the IPv4 address space of only 4 octets. OSI networks have not been well accepted, however, and the OSI protocol stack has been relegated to the status of OSI Reference Model. See also *datagram*, *error control*, *fragmentation*, *IP*, *ISO*, *lifetime control*, *Network Layer*, *OSI*, *OSI Reference Model*, *protocol*, and *protocol stack*.

**clocking pulse** Periodic signals generated by a timing source for purposes of synchronizing the flow of data within a computer or between computers across a circuit. See also *synchronous transmission*.

**closed circuit television (CCTV)** See *CCTV*.

**closed-loop algorithm** In frame relay, a congestion control mechanism that prevents the frame relay network device (FRND) from accepting incoming frames unless there is an extremely high probability of the network's being able to deliver them without discard. A closed-loop algorithm fairly allocates backbone bandwidth among all the permanent virtual circuits (PVCs) configured on a particular trunk, and in proportion to the Committed Information Rate (CIR) of each PVC. See also *backbone*, *bandwidth*, *CIR*, *congestion*, *frame relay*, *FRND*, *PVC*, and *trunk*.

**closed user group (CUG)** See *CUG*.

**cloud** A wide area network (WAN) commonly is depicted as a cloud, which serves to obscure its complex inner workings from view. Data just pops in on one side of the cloud and pops out on the other side, so to speak.

**LAN (Local Area Network)** A LAN is a packet network designed to interconnect host computers, peripherals, storage devices, and other computing resources within a local area, i.e., limited distance. LANs conform to the client/server architecture, a distributed computing architecture that runs applications on client microcomputers against one or more centralized servers, which are high-performance multiport computers with substantial processing power and large amounts of memory. A LAN might serve an office, a floor of a building, and entire building, or a campus area, but generally does not cross a public right-of-way such as a street. The distance limitation generally is in the range of a few kilometers, at most, although that is sensitive to the transmission media employed, which include coaxial cable, twisted pair, optical fiber, infrared (IR) light, and radio frequency (RF) systems. Raw bandwidth ranges up to 10 Gbps, although actual throughput generally is much less. LANs generally are private networks, although public wireless hotspots offering wireless Internet access currently are popular. Most LAN standards are set by the 802 Working Group of the Institute of Electrical and Electronic Engineers (IEEE), with examples being 802.3 (Ethernet) and 802.11a/b/g (Wi-Fi). A personal area network (PAN) such as Bluetooth, is much more limited in geographic scope than a LAN. LANs and LAN segments can be interconnected over a metropolitan area network (MAN) or wide area network (WAN). LANs operate at Layer 1, the Physical Layer, and Layer 2, the Data Link Layer, of the OSI Reference Model. See also *802.3*, *802.5*, *802.11*, *architecture*, *bandwidth*, *Bluetooth*, *client/server*, *coaxial cable*, *Data Link Layer*, *Ethernet*, *hotspot*, *IEEE*, *IR*, *MAN*, *optical fiber*, *OSI Reference Model*, *PAN*, *Physical Layer*, *RF*, *throughput*, *Token Ring*, *twisted pair*, *WAN*, and *Wi-Fi*.

**landline** **1.** A traditional telephone connected to the PSTN by a traditional wire (or fiber) local loop that terminates in a fixed location, rather than a cellular mobile telephone connected to a cellular network via radio technology. A cordless telephone is considered part of a landline as the local loop terminates in a fixed base station on the subscriber premises, even though the connection to the base station is wireless. A wireless local loop (WLL) is considered a landline, as it is terrestrial and connects two fixed points. See also *local loop* and *WLL*. **2.** A telecommunications system that uses traditional terrestrial cabled, or conducted, transmission media such as copper or fiber optics, and wireless systems such as microwave, rather than mobile wireless radio technologies such as cellular or, especially, non-terrestrial satellite.

**LANE (LAN Emulation)** A specification (January 1995) from the ATM Forum (since merged into the MFA Forum) for an ATM service in support of native Ethernet (802.3) and Token Ring (802.5) local area network (LAN) communications over an ATM network. Software in the end systems (e.g., ATM-based hosts or routers, known as *proxies*), of the ATM network emulates a native LAN environment. LANE acts as Layer 2 bridge in support of connectionless LAN traffic, with the connection-oriented ATM service being transparent to the user application. In LANE, a LAN emulation client (LEC) connects to the ATM network over a LANE user-to-network interface (LUNI). The network-based LAN emulation server (LES) registers the LAN medium access control (MAC) addresses and translates them into ATM addresses using the address resolution protocol (ARP). Each LEC is assigned to an emulated LAN (ELAN) by an optional network-based LAN emulation configuration server (LECS). Each LEC also is associated with a broadcast and unknown server (BUS) that handles broadcast and multicast traffic, as well as initial unicast frames before address resolution. LANE traffic generally is Class C variable bit rate (VBR) traffic in message mode, and is supported over ATM Adaptation Layer Type 5 (AAL5). See also *802.3*, *802.5*, *AAL5*, *ARP*, *ATM*, *ATM Forum*, *broadcast*, *BUS*, *Class C ATM traffic*, *connectionless*, *connection-oriented*, *ELAN*, *emulation*, *Ethernet*, *host*, *Layer 2*, *LEC*, *LECS*, *LES*, *LUNI*, *MAC*, *message mode service*, *MFA Forum*, *multicast*, *proxy*, *router*, *Token Ring*, *unicast*, and *VBR*.

**LAN emulation (LANE)** See *LANE*.

**LAN emulation client (LEC)** See *LEC*.

**LAN emulation configuration server (LECS)** See *LECS*.

**LAN emulation server (LES)** See *LES*.

**LANE user-to-network interface (LUNI)** See *LUNI*.

# Appendix G

# PART 1

# Packet Network Foundations

Part I of this book introduces four widely deployed packet network technologies: X.25, frame relay, asynchronous transfer mode (ATM), and Internet protocol (IP).

Before packet networks, communications technology used circuit-switched telephone networks with dedicated, analog circuits that functioned on a "always on once activated" basis. A dedicated circuit cannot be used for other purposes even if no communications are taking place at the moment. In regard to telephone conversations, it is estimated that on the average a dedicated circuit carried active traffic only 20 to 25 percent of the time and is idle the other 75 to 80 percent. Moreover, other services such as video data streams cannot be efficiently carried on circuit-switched networks.

Packet networks based on packet switching technologies represent a radical departure. The key idea behind packet switching is that a message or a conversation is broken into independent, small pieces of information called *packets* that are either equal or variable in size. These packets are sent individually to a destination and are reassembled there. No physical resource is dedicated to a connection, and connections become virtual, thus allowing many users to share the same physical network resource.

The concept of packet switching is attributed to Paul Baran who first outlined its principles in an essay published in 1964 in the journal *On Distributed Communications*. The term *packet switching* itself was coined by Donald Davies, a physicist at the British National Physical Lab, who came up with the same packet switching idea independently. It is interesting to note that a few decades earlier, a similar discovery in physics by Albert Einstein—that waves of light can be broken into a stream of individual photons—led to the development of quantum mechanics.

Packet networks allow more efficient use of network resources. Each packet occupies a transmission facility only for the duration of the transmission, leaving the facility available for other users when no transmission is taking place.

Packet-switched networks are highly fault-tolerant. From the very start of their development, network survivability was a major design goal. Because packet networks do not rely on dedicated physical connections, packets can be routed via alternative routes in case of an outage in the original communications link.

Packet networks can support bandwidth on-demand and flexible bandwidth allocation. Bandwidth is allocated at the time of communication, and the amount of bandwidth allocated is based on need. In

CHAPTER 8

# Local Area Networks

# 8.1 Introduction

A local area network is a high-speed data network that covers a relatively small geographic area. It typically connects workstations, personal computers, printers, servers, and other end-user devices, which are collectively also known as *data terminal equipment*. The common applications of LAN include shared access to devices and applications, file exchange between connected users, and communication between users via electronic mail and others. LANs are also private data networks, because they belong to an organization and are used to carry data traffic as opposed to voice traffic.

This section provides a brief introduction to LAN history, standards, protocol stacks, topologies, and devices.

## 8.1.1 LAN History and Standards

LAN is a type of broadband packet access network that carries the packet data traffic of an organization. LAN interconnects the end users of an organization to an outside public data network such as the Internet.

The basis of LAN technologies and standards was defined in the late 1970s and early 1980s. LAN technologies really emerged with the Internet itself, and the first widely deployed LAN technology, Ethernet, is almost as old as the Internet itself. The overwhelming majority of the deployed LANs are Ethernet.

IEEE 802, a branch of the International Institute of Electrical and Electronics Engineers (IEEE), is responsible for most of the LAN standards. These standards have also been adopted by other standards organization such as ANSI and ISO. The major LAN standards are listed in Table 8-1.

## 8.1.2 LAN Protocol Stacks

The LAN protocols operate at the bottom two layers of the OSI network reference model, i.e., at the physical layer and the data link layer, as shown in Fig. 8-1. The physical layer is primarily concerned with the transmission medium and its physical characteristics for digital signal transmission. The data link layer consists of two sublayers, the medium access control (MAC) sublayer and logical link control (LLC) layer. The MAC sublayer is responsible for controlling access to a shared medium by multiple users simultaneously. The LLC sublayer is responsible for

**TABLE 8-1**

IEEE 802 LAN Standards Summary

| IEEE 802 specification | LAN technology | Description |
|---|---|---|
| IEEE 802.1 (ISO 15802-2) | General information | Details how the other 802 standards relate to one another and to the ISO OSI reference model. |
| IEEE 802.2 (ISO 8802.2) | LLC framework | Divides the OSI data link layer into two sublayers and defines the functions of the LLC and MAC sublayers |
| IEEE 802.3 | Ethernet | Defines the CSMA/CD protocol, which is used in Ethernet applications and has become synonymous with Ethernet |
| IEEE 802.4 (ISO 8802.4) | Token bus | Defines the token-passing bus access method |
| IEEE 802.5 (ISO 8802.5) | Token ring | Defines the Token Ring access method |
| IEEE 802.7 | Broadband LAN | Recommended practices for broadband LANs |
| IEEE 802.11 | Wireless LAN | Wireless LAN medium access control (MAC) and physical layer specifications |
| IEEE 802.15 | Wireless personal area network (WPAN) | WPAN MAC and physical layer specifications |
| IEEE 802.16 | Broadband fixed wireless metropolitan area networks (MANs) | Air interface specification for fixed broadband wireless access systems |
| IEEE 802.12 | 100 VG-AnyLAN | Defines a LAN technology that supports the operations of any existing LAN protocol, including the Ethernet frame format and Token Ring frame format, but not both at the same time |

interfacing to the upper layers, such as IP and the Internetwork Packet Exchange protocol (IPX). Any layer above the data link layer is beyond the scope of the LAN protocols.

The IEEE 802 LAN standards are compatible at the upper part of the data link layer, i.e., at the LLC sublayer, but differ from each other at the MAC sublayer and physical layer.

The scope of each LAN protocol may vary. Some cover the entire two bottom layers. For example, Ethernet as defined in IEEE 802.3 (IEEE 2001) and FDDI as defined in IEEE 802.5j (IEEE 1998c) cover the physical layer and both sublayers of the data link layer, as shown on the right-hand side of Fig. 8-1. Other LAN protocols, such as Token Ring and token bus,

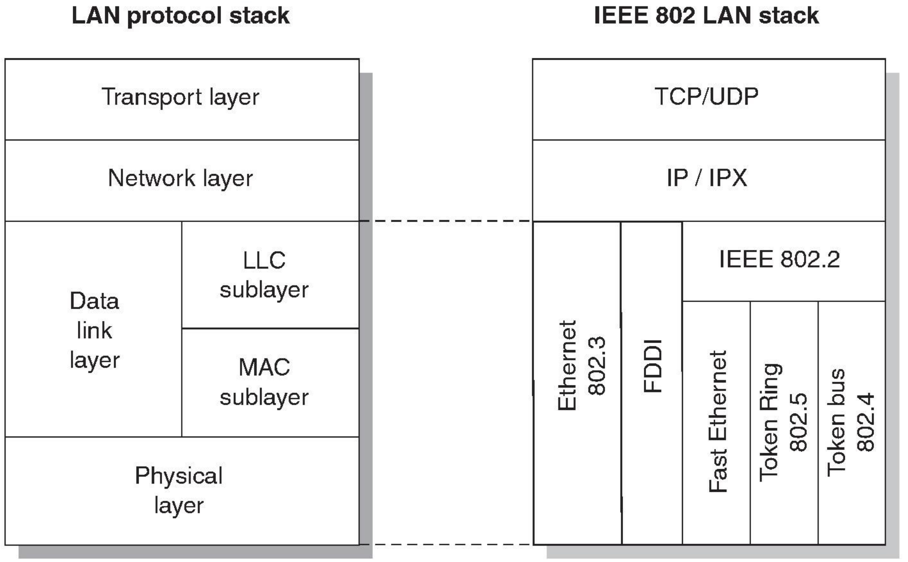



**Figure 8-1** The LAN protocol stack.

specify the physical layer and the MAC sublayer while sharing a common LLC specification defined in IEEE 802.2, as shown on the right-hand side of Fig. 8-1.

**8.1.2.1 Physical Transmission Medium** The LAN transmission medium can be divided into the two general categories of wired and wireless. This chapter focuses only on the wired or wireline LAN technology, while Chap. 9 will describe wireless LAN.

There are basically three types of transmission media used in wireline LAN deployment: copper twisted pair, coaxial cable, and optical fiber. The type of transmission medium determines the data rate and transmission distance.

**TWISTED PAIR COPPER WIRE** Twisted pair, both shielded and unshielded, is a pair of copper wires that are twisted to increase the transmission distance. It is the least costly of the three wireline LAN media, and one of the most common transmission media currently used in LAN applications. It is primarily used in star and hub LAN configurations in office buildings. The maximum transmission distance of twisted pair cable depends on the target date rate; typically the limit is 100 m without repeater. The data rate of copper twisted pair normally is not as high as that of other transmission media and depends on factors such as transmission distance and the modulation scheme used for transmission. The

longer the transmission distance is, the lower the bit rate is. It is not uncommon to see twisted pair achieve a bit rate of over 1 Mbps for a distance of 100 meters.

**COAXIAL CABLE** Coaxial cable, whose transmission wire is insulated with dielectric insulating material and braided out conductor, can achieve higher data rates and longer transmission distances. There are two kinds of coaxial cables: thin wire and thick wire, referring to the difference in the cable diameters, thin wire being 0.25 in diameter and thick wire being 0.5 in diameter. Thin-wire coaxial cable reaches shorter distances, typically 200 m with the data rate of over 10 Mbps, while thick-wire cable can reach over 500 m with the same data rate.

**OPTICAL FIBER** Optical fiber carries data in the form of flashing light beams in a glass fiber, as opposed to electrical signals on a wire. Optical fiber can achieve much higher data rates than coaxial cable or twisted pair over much longer distances. The fiber transmission equipment consists of fiber cable, special electrical-to-optical and optical-to-electrical converters, light emitters such as light-emitting diodes (LEDs) or laser and optical receivers. These transmission components have been much more costly than twisted pair and coaxial cable. However, with the advent of new optical transmission technologies and a massive market for broadband applications, the cost has come down considerably in recent years and the optical fiber is becoming a common choice for LAN deployment.

LANs can use one type of transmission medium or a mix of types. For example, lower-speed twisted pair can be used between a computer and a hub, while coaxial cable can be used between a branch hub and a main hub and high-speed optical fiber cable can be used between a main hub and an outside router.

**8.1.2.2 Media Access Control Sublayer** A LAN technology must address the issue of resource contention because multiple users share the same transmission medium. A contention occurs when two DTEs transmit data at the same time. There are basically two MAC mechanisms for LAN: carrier-sense multiple access with collision detection and control token.

**CSMA/CD** The CSMA/CD access control method is used in Ethernet and can be characterized as "listen and send." A network device first listens to the wire when it has data to send, then sends the data when it finds that no other device is sending the data. After it finishes sending the

data, it listens to the wire again to detect if any collision occurs while it transmitted data. A collision occurs when two devices send data simultaneously. If a collision is detected, the device waits for a random amount of time before resending the data. The randomness of the wait period makes the possibility of another collision very small. However, this algorithm is not deterministic, and when the number of users increases to a large enough point, network performance deteriorates drastically owing to the large number of collisions.

The major advantage of CSMA/CD is its simplicity. It is easy to implement and works well in the LAN environment.

**CONTROL TOKEN** Control token is a special network packet used to control access to a shared transmission medium. A token is passed around a network from device to device. When a device has data to send, it must wait until it has the token, at which time it sends its data. When the transmission is complete, the token is released so that other devices may use the network to transmit their data. A major advantage of token-passing networks is that they are deterministic. In other words, it is easy to calculate the maximum time that will pass before a device has the opportunity to send data. This explains the popularity of token-passing networks in some real-time environments such as factories, where machinery must be capable of communicating at determinable intervals. Token-passing networks include Token Ring and FDDI.

**MAC ADDRESS** The MAC address is a number that is hard-wired into each LAN card such as the Ethernet Network Interface Card or adapter that uniquely identifies this device on a LAN. The MAC addresses are 6 bytes in length, and are usually written in hexadecimal such as 12:34:56:78:90:AB. The colons in the address may be omitted, but generally make the address more readable. Each manufacturer of LAN devices has a certain range of MAC addresses, just like a range of telephone numbers, that they can use. The first 3 bytes of the address denote the manufacturer.

**8.1.2.3 Link Layer Control Sublayer** The LLC sublayer, as defined in the IEEE 802.2 standard, mainly hides the differences between various MAC sublayer implementations such as Ethernet, Token Ring, and FDDI and presents a uniform interface to the network layer. This allows different types of LANs to communicate with each other.

The IEEE 802 LLC protocol defines a generic LLC protocol data unit that includes both user data and LLC header. The LLC header contains a control field that in turn contains the fields such as protocol ID and

header type. Also found in the LLC header are source and destination address fields.

## 8.1.3 Data Transmission Methods

There are three data transmission modes in LAN environments: point-to-point, multicast, and broadcast. In each transmission mode, a single packet is sent to one or more nodes.

In *point-to-point transmission,* which is also known as *unicast,* a single packet is sent from a source to a destination on a LAN. First, the source node addresses the packet by using the address of the destination node. The packet is then sent onto the LAN, and the LAN then passes the packet to its destination.

In *multicast transmission,* a single data packet is copied and sent to a specific subset of nodes on a LAN. First, the source node addresses the packet by using a special type of address, called a *multicast address.* The packet is then sent onto the LAN, which makes copies of the packet and sends a copy to each node that is part of the multicast address.

In *broadcast transmission,* a single piece of data is copied and sent to all the nodes on a LAN. In this type of transmission mode, a source node addresses a packet by using a broadcast address. The packet is then sent onto the LAN, which makes copies of the packet and sends a copy to every node on the LAN.

## 8.1.4 LAN Topology

A LAN topology defines how the data terminal equipment such as desk top computers, printers, and server computers, and LAN internetworking devices such as switches, routers, and hubs, are interconnected to each other. In general, there are four types of LAN topologies: bus, star, ring, and hub. Each has some advantages, which will now be discussed (Halsall 1996).

**8.1.4.1 Bus Topology** The bus is one of the most common LAN topologies. A simple bus topology is characterized by a central cable that runs through end-user equipment like computers and servers, as shown in Fig. 8-2(a). A physical connection, also known as a *tap,* is made to the cable for each user terminal or computer to access the network. MAC circuitry and the software implementing the control scheme together allow the connected users to share the common cable and transmission

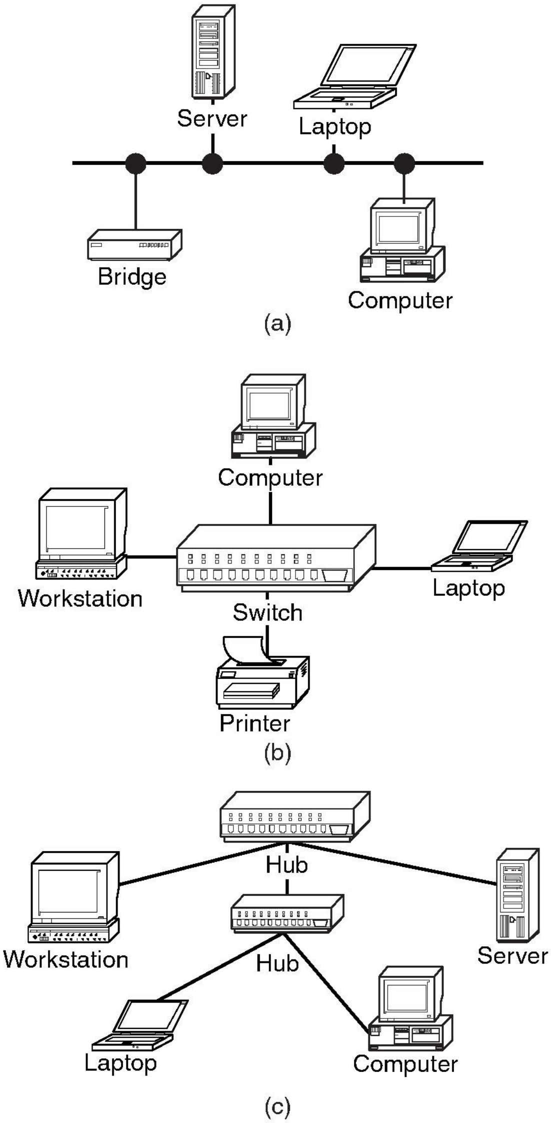



**Figure 8-2** Bus, star, and hub LAN topology examples.

bandwidth. A slightly more complicated bus topology may consist of multiple layers of buses. A bus cable can be connected to another bus cable, which in turn may be connected to yet another cable. This forms a topology that looks like an uprooted tree.

**8.1.4.2 Ring Topology** Ring-based LAN topology is characterized by a cable that passes from one DTE to another until all the DTEs are connected to form a ring or loop. A distinct feature of ring topology is

that traffic travels in one direction only. Between two neighboring user DTEs, it is a direct point-to-point link that carries traffic in one direction only, termed *unidirectional*. Again, medium access circuitry and a control algorithm are built into a DTE and network to allow each DTE a fair chance to access the cable ring.

**8.1.4.3 Star Topology** In star topology, there is a focal point that is either a switch or a router, and all end-user DTEs are connected to the central point via a point-to-point cable, as shown in Fig. 8-2(b). This is a typical voice-service PBX configuration that is also used to interconnect end-user DTEs, although not as common as other topologies. Compared to the other topologies, star topology has more complicated wiring.

**8.1.4.4 Hub Topology** A fourth common topology is the hub structure, which is a mix of the ring and bus topologies. A hub topology is simply a bus or ring wiring collapsed into a central unit. A hub does not perform any switching or intelligent processing. All a hub does is simply retransmit all signals received from a DTE to all other DTEs with a set of repeaters inside the hub. As shown in Figure 8-2(c), a hub can be connected to another hub to form a hierarchy of hubs and DTEs that looks like a tree structure.

## 8.1.5 LAN Internetworking Devices

An internetworking device interconnects two or more other LAN devices. Based on functionality, there are three types of such internetworking devices: repeater, bridge, and router or switch.

**8.1.5.1 Repeater** A LAN *repeater* is a physical layer device used to connect two LAN cable segments so a LAN will extend further in distance. A repeater essentially boots digital signals to allow a series of cable segments to be treated as a single cable. It receives signals from one network segment and amplifies, retimes, and retransmits those signals to another network segment. A LAN repeater operates at the physical layer without any intelligence to perform any filtering or other traffic processing. In addition, all electrical signals, including electrical noise and errors, are repeated and amplified as well. The total number of repeaters within a LAN is limited due to timing and other issues.

**8.1.5.2 LAN Hub** A *hub* is a physical layer device that connects multiple user stations, each through a dedicated cable. In some respects, a

hub functions as a multiport repeater. Hubs are used to create a physical star network while maintaining the logical bus or ring configuration of a LAN.

**8.1.5.3 LAN Bridge** A LAN *bridge* is an internetworking device that interconnects two LAN segments at the data link layer as opposed to the physical layer in the case of a repeater. A bridge must have at least two ports, one receiving incoming frames and one sending outgoing frames. A bridge uses a MAC address to route frames from one segment to another, or even to a different LAN that is the same or different at the physical or MAC layer.

**8.1.5.4 LAN Router and Switch** A LAN *router* operates at the network layer, interconnecting like and unlike devices attached to one or more LANs. LAN routers normally also support link layer bridging in addition to network layer routing.

A LAN router, as described in Chap. 4 on IP networks and Appendix A, employs routing protocols to dynamically obtain knowledge of destination address prefixes across an entire set of internetworked LANs. A LAN router normally has a packet-forwarding engine that uses a lookup table to identify the physical interface of the next hop toward the destination.

## 8.2 Ethernet

Ethernet is almost as old as the Internet itself. Since its inception at a Xerox lab in the early 1970s, it has been the dominant protocol for local area networks. By various estimates, Ethernet accounts for somewhere between 80 to 95 percent of worldwide LAN installations.

This section, after first providing some background information, introduces three generations of Ethernet: 10Base Ethernet, Fast Ethernet, and optical Ethernet, with an emphasis on the first two. Gigabit Ethernet and 10 Gigabit Ethernet were described in detail in Chap. 7 in the context of optical transport network, and will be discussed briefly in this chapter in the context of LAN technology.

What is remarkable about Ethernet is its continuity and simplicity. The fundamentals of Ethernet such as Ethernet frame and logical link control, which were already defined for the first generation of Ethernet, have remained largely intact through the rapid technological evolution

of the past two decades. Ethernet is viewed as a kind of plug-play technology because it is relatively simple and can operate with very little manual intervention for configuration and provisioning.

## 8.2.1 Ethernet Basics

**8.2.1.1 A Brief History** Ethernet was originally developed by Digital, Intel, and Xerox (DIX) in 1972 and was designed as a "broadcast" system where stations on a network can send messages at will. All the stations may receive the messages, but only one specific station to which the message is directed will respond. Robert Metcalf and David Boggs of Xerox are credited with coming up with first Ethernet design. Ethernet was originally designed to run on any medium (copper wire, fiber, or even radio wave), which is where *Ether* in the term *Ethernet* comes from.

The original version of Ethernet was adopted by IEEE Committee 802.3 (IEEE Project 802 was named after the time Ethernet was set up, in February 1980), and the packet format was standardized, which is known as the IEEE 802.3 Ethernet frame.

The Ethernet evolution, based on the transmission technologies and speed, involved at various times in its development, can be divided into the following periods:

- 10BaseT Ethernet, starting from 1972 to the mid-1990s
- 100Base Ethernet, starting from the mid-1990s
- 1000Base Ethernet, starting from 1998
- 10Gig Ethernet, starting from 2000

An Ethernet version is represented in terms of the transmission speed, the transmission medium, and maybe the transmission distance. The prefix number in an Ethernet version such as 10 in 10Base or 100 in 100Base refers to the transmission speed of 10 Mbps and 100 Mbps. The suffix letter refers to the medium type, while suffix number for earlier versions of Ethernet refers to the maximum transmission distance. For example, the letter T in 10BaseT refers to "twisted pair" copper wire and the number 5 in 10Base5 refers to the transmission distance in hundreds of meters.

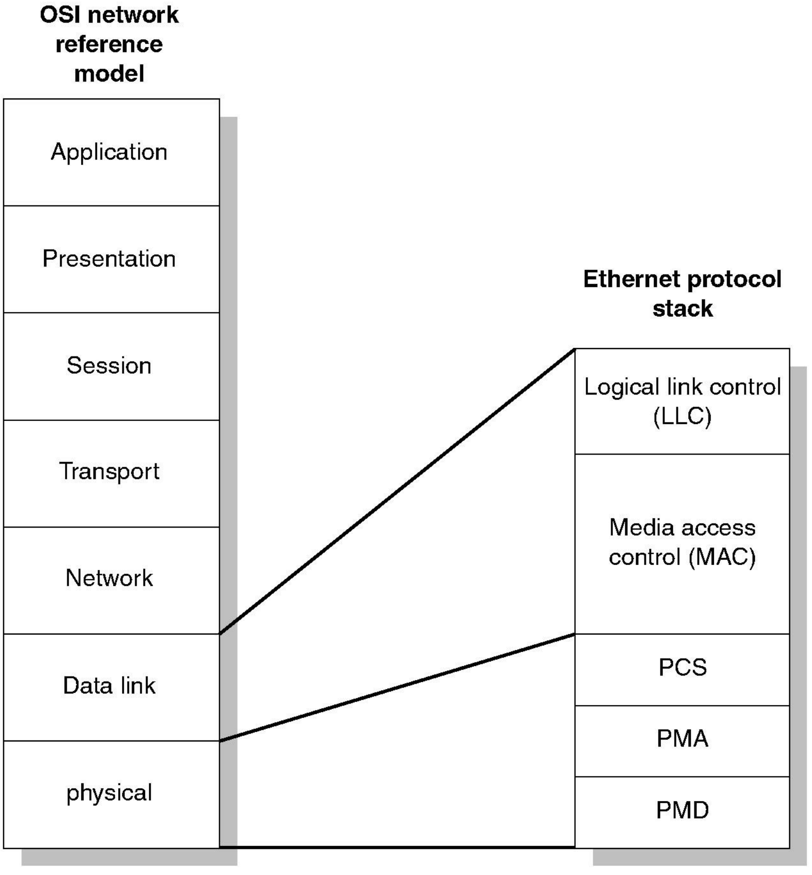
**8.2.1.2 Ethernet Protocol Stack** The Ethernet protocol stack is similar to the general LAN protocol stack as described earlier: It covers the layers 1 and 2 of the OSI network reference model. In addition, Ethernet further

defines three sublayers for the physical (PHY) layer: PMD, PMA, and PCS, which are briefly discussed here (IEEE 2001c).

The physical medium-dependent (PMD) sublayer defines the Ethernet cables, wiring, and other transmission medium-related components. The physical medium attachment (PMA) defines the type of connectors used to connect an Ethernet device such as an Ethernet NIC, hub, or switch to the Ethernet cable. The physical coding sublayer (PCS) defines a scheme appropriate to the medium to encode and decode data received from/sent to the PMD sublayer (Spurgen 2000).

The Ethernet data link layer, like that of other LAN technologies, is broken into two sublayers: the LLC on the upper half and the MAC on the lower half. The MAC deals with getting data on and off the wire and media access control, as shown in Fig. 8-3. The logical link control

**Figure 8-3**
The Ethernet protocol stack in reference to the OSI network reference model.



on the upper half of the data link layer deals with error checking and providing a uniform interface to the network layer above.

**8.2.1.3 Ethernet Operation Mode** Ethernet supports either half-duplex, full-duplex, or both operation modes. Early Ethernet supports only the half-duplex mode of operation, where a station can transmit or receive data but not at the same time. In contrast, a station supporting the full-duplex mode of operation can transmit and receive data simultaneously. It was with the development of Fast Ethernet that Ethernet became able to support both half-duplex and full-duplex modes of operation.

## 8.2.2 First Generation—10BaseT Ethernet

10BaseT is one of the most popular versions of the first generation of Ethernet, and defines the fundamentals of Ethernet technology upon which later generations of Ethernet have been built. This discussion will cover the area of the physical layer, the media access control sublayer, and the logical link control sublayer.

**8.2.2.1 Physical Layer of Ethernet** The characteristics of the first generation of Ethernet are summarized in Table 8-2, which includes the transmission medium, transmission distance and data rate, and operation mode.

**TABLE 8-2**
10Base Ethernet Summary

| Standards | IEEE standard—year first released | PMD type | Date rate | Maximal distance in meters: Half duplex | Maximal distance in meters: Full duplex |
|---|---|---|---|---|---|
| 10Base5 | 802.3—1983 | Coax cable (thick Ethernet) | 10 Mbps | 500 | Not supported yet |
| 10Base2 | 802.3—1985 | Coaxial cable (thin Ethernet) | 10 Mbps | 185 | Not supported yet |
| 1Base5 | 802.3—1987 | 2 pairs of twisted telephone cable | 1 Mbps | 250 | Not supported yet |
| 10Base-T | 802.3—1990 | 2 pairs of category 3 or better UTP cable | 10 Mbps | 100 | 100 |
| 10Base-FL | 802.3—1993 | Two optical fibers | 10 Mbps | 2000 | >2000 |

SOURCE: IEEE 2000.

The transmission medium has evolved from the original thick coax (10base5) to twisted pair copper wire and then to fiber. Twisted pair is the most common choice of cable for the first generation of Ethernet. Unshielded twisted pair (UTP) is one kind of twisted pair that has two copper wires twisted together and is relatively immune to noise.

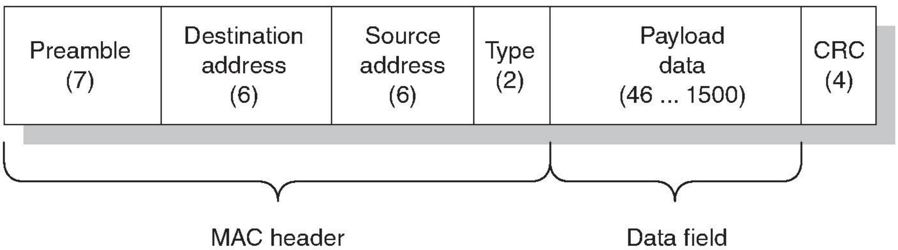
The physical coding sublayer uses Manchester coding, a common coding scheme at the time of first-generation Ethernet that divides each bit into two halves. A 1 is defined by a transition from "low" to "high" in the middle of the bit period, and a 0 is defined as a transition from "high" to "low" in the middle of the bit period.

Most versions of first-generation Ethernet support only the half-duplex mode of operation and have a transmission distance of around 250 m.

**8.2.2.2 Ethernet Frame** The Ethernet frame defines a structure to hold user data and to be carried on the physical medium. It consists of two parts: a header and the payload data. Figure 8-4 shows the IEEE 802.3 Ethernet frame format, which includes the following:

- *Preamble.* A 7-byte field containing a series of alternating 1s and 0s used by an Ethernet receiver to acquire bit synchronization and frame timing information. This field is generated by the hardware in an Ethernet device.
- *Destination address.* The MAC address of a receiving Ethernet device.
- *Source address.* The MAC address of a sending device.
- *Type.* A 2-byte field indicating the type of data encapsulated, e.g., IP, ARP, RARP, etc.
- *Payload data.* The data field with length ranging from 46 to a maximum of 1500 bytes.
- *Cyclical redundancy check (CRC).* A 4 -byte field used for error detection.

**Figure 8-4**
The IEEE 802 Ethernet frame structure.



**8.2.2.3 Media Access Control** Ethernet MAC uses CSMA/CD for access control. By means of carrier sense multiple access, with collision detection, an Ethernet device does the following:

1. Listens to the line before putting a packet "on the wire," and if the line is busy, waits for a predetermined number of seconds before retry
2. When the line becomes idle, transmits while monitoring for collisions
3. If a collision is detected, sends the jam signal and waits for an algorithmically determined number of seconds before resending any packets
4. If the maximum number of transmission attempts is reached, gives up

## 8.2.3 Second Generation—Fast Ethernet

Fast Ethernet was defined to meet the demands of fast-growing Internet traffic. In the face of fast growth, 10 Base Ethernet became too slow by the early 1990s to meet all the needs of the Internet's data traffic flow. The IEEE reconvened the IEEE 802.3 committee in 1992 to upgrade Ethernet to 100 Mbps. Two competing proposals emerged in the process: one simply aimed at increasing the speed of the existing Ethernet as defined by IEEE 802.3 to 100 Mbps, while the other reworked the old Ethernet with a new architecture. The first proposal resulted in the updated IEEE 802.3 specifications, also known as *Fast Ethernet,* that were approved in 1996. The second resulted in the establishment of the IEEE 802.12 committee and the 802.12 standard specifications in 1995, also known as *100VG-AnyLAN.* This subsection briefly describes Fast Ethernet, while the following subsection discusses 100VG-AnyLAN.

One major change in the Fast Ethernet specifications is that shared medium topologies like the bus topology are eliminated in favor of the star topology in order to decrease transmission collisions and increase network throughput. At the center of the star topology is a switching hub that supports full-duplex operation.

**8.2.3.1 Physical Layer** The Fast Ethernet specifications define three physical media, or physical medium-dependents: 100Base-T4, 100BaseSE-TX, and 100Base-FX. The 100Base-T4 uses four unshielded

twisted pairs of cable to connect a user station to a hub, a very common situation in office buildings. The 100Base-TX uses two pairs of category 5 unshielded twisted pairs. The 100Base-FX uses a pair of optical fiber cables that are defined by ANSI standards for FDDI. Table 8-3 summarizes the Fast Ethernet physical layer characteristics.

Fast Ethernet adopts a faster coding scheme at the physical signaling sublayer, i.e., the 4-bit/5-bit scheme that uses groups of four data bits as a transmission unit, also called an *encoded symbol*, with the fifth bit as the delimiter.

**8.2.3.2 MAC Layer** Fast Ethernet retains the original Ethernet MAC layer. All the original frame formats, procedures, and media access control algorithms, i.e., CSMA/CD, remain almost identical. This enables the first-generation of 10-Mbps Ethernet LANs to run over 100 Mbps Fast Ethernet without any changes.

## 8.2.4 100VG-AnyLAN

The IEEE 802.12 standards, originally approved in 1995, were the result of a competing proposal for upgrading the first generation of Ethernet. The central idea behind 100 VG-AnyLAN is to define a LAN technology that supports the operations of any existing LAN protocol, be it Ethernet frame format and Token Ring frame format, but not both at the same time. The main goals of 100VG-AnyLAN include avoiding the frame collisions of the traditional CSMA/CD access method and providing

**TABLE 8-3**

100Base Ethernet Summary

| Standards | IEEE standard—year first released | PMD type | Maximal distance in meters: Half duplex | Maximal distance in meters: Full duplex |
|---|---|---|---|---|
| 100Base-TX | 802.3—1995 | Two pairs of category 5 UTP cable | 100 | 100 |
| 100Base-FX | 802.3—1995 | Two optical fibers | 400 | 2000 |
| 100Base-T4 | 802.3—1995 | Four pairs of category 3 or better UTP cable | 100 | Not supported |
| 100Base-T2 | 802.3—1997 | Two pairs of category 3 or better UTP cable | 100 | 100 |

SOURCE: IEEE 2000b.

prioritized services on LAN (IEEE 1998a). 100VG-AnyLAN did not achieve wide acceptance in the market, largely due to the overwhelming dominance of Ethernet.

The prioritized service is implemented via a demand priority protocol that utilizes a round robin polling scheme for each station to request a priority for each MAC frame from the repeater. Higher priority is given to delay-sensitive frames, while the best-effort service is given to the rest of the frames.

Collision avoidance is achieved via the exclusive use of a switching hub as opposed to the shared media used by traditional Ethernet. A station can transmit only after it is granted permission to do so by the connected repeater. Thus the access control method is deterministic with no collisions.

## 8.2.5 Gigabit and 10 Gigabit Ethernet

Gigabit Ethernet and 10 Gigabit Ethernet as transport technologies are introduced in Chap. 7 on optical ethernet, which focuses on the physical layer of both Ethernet technologies. This subsection provides an overview of Gigabit Ethernet and 10 Gigabit Ethernet from the perspective of LAN.

**8.2.5.1 Gigabit Ethernet** Soon after the Fast Ethernet standards were finalized, the work on 1000Base Ethernet began at the IEEE 802.3z committee. After the specifications were completed, large-scale deployment soon followed.

One primary goal of Gigabit Ethernet, like its predecessor Fast Ethernet, is to alleviate the bandwidth crunch on LANs with 10-fold increase in speed. Gigabit Ethernet also preserves the standard 802.3 Ethernet frame format and the minimum and maximum sizes of the frame, so that it is backward-compatible with 100BaseT and 10BaseT Ethernet.

Gigabit Ethernet supports both full- and half-duplex operations, the same as Fast Ethernet. For half-duplex operations, CSMA/CD is used. For full-duplex operations, the standard flow control defined in IEEE 802.3 is used (IEEE 2001b). At the physical layer, Gigabit Ethernet supports both fiber and copper wire as physical media, although optical fiber is the common choice. It uses the recently defined ANSI Fibre Channel standards as the basis for fiber-based media (ANSI 1998).

Gigabit Ethernet equipment, like Ethernet switch or router equipment, is mainly used for LAN backbone, interconnecting distributed multiple LANs, or connecting a LAN to a backbone IP network.

**8.2.5.2 10 Gigabit Ethernet** Efforts on the 10 Gigabit Ethernet specifications by the IEEE 802.ae committee were initiated soon after the Gigabit Ethernet specifications were completed. The 10 Gigabit technology clearly targets LAN, the traditional space of Ethernet, and the space beyond LAN such as WAN and MAN. 10 Gigabit Ethernet defines two families of physical layer interfaces: one for local area networks, operating at a data rate of 10 Gbps, and one for wide area networks, operating at a data rate compatible with the payload rate of OC-192c/SDH VC-4-64c. 10 Gigabit Ethernet preserves the standard 802.3 Ethernet frame format and the minimum and maximum sizes of the frame, so that it is backward-compatible with 100BaseT and 10BaseT Ethernet, like Gigabit Ethernet.

One important feature of 10 Gigabit Ethernet is that it supports full-duplex operation only. The traditional Ethernet half-duplex operation for shared connections and CAMA/CD is abandoned.

# 8.3 Token Ring LAN

Token Ring LAN technology was originally developed by IBM in the 1970s, was originally standardized by the IEEE as the standard IEEE 802.5 in 1985, and then was adopted as ISO 8802.5 (IEEE 1998c). The IEEE 802.5 specification is almost identical to IBM's Token Ring network, with some minor differences. Throughout this chapter, the term *Token Ring* generally is used to refer to both IBM's Token Ring network and IEEE 802.5 network unless noted otherwise.

The Token Ring network is well suited for use in commercial and industrial environments, where predictability of the performance is expected.

## 8.3.1 Transmission Medium

IBM Token Ring uses twisted pair copper wire as the transmission medium even though IEEE 802.5 does not specify a media type. In more recent deployments, optical fiber cable is also used to extend the size of the ring interconnecting hubs beyond their normal limitations.

With unshielded twisted pair, a very common wiring choice, a Token Ring network can have a maximum of 72 stations or nodes, although in practice the number of nodes is normally smaller. With shielded twisted pair (STP) wiring, the number of attached stations can increase up to 250

in theory. The typical distance of a Token Ring LAN, called an *average ring length* (ARL), is about 100 m, and this distance can be extended 10-fold if optical fiber cable is used between hubs.

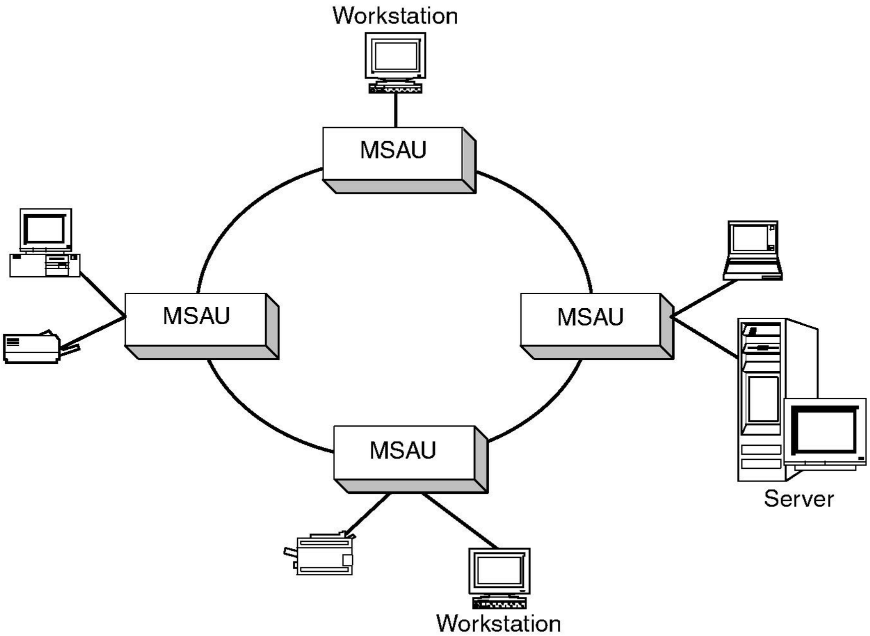
The original IEEE 802.5 Token Ring LAN operates at 4 Mbps, but the standard now covers transmission rates up to 16 Mbps.

## 8.3.2 Token Ring LAN Configuration and Topology

A Token Ring network typically features a ring topology formed from a set of small clusters or stars, as shown in Fig. 8-5. At the center of each star is a multistation access unit (MSAU) with a set of Token Ring stations connected to it. An MSAU is basically a hub device, and each station is connected to it via a twisted pair cable with two wire pairs. One pair receives data and the other is for transmitting data. The MSAUs are connected together with patch cable or optical fiber cable to form a ring.

An MSAU can be passive or active. A passive MSAU merely provides an electrical path for the data to pass through. An active MSAU amplifies the signals passing through it. With active MSAUs, a Token Ring network can extend further in distance.

**Figure 8-5**
A Token Ring network topology.



## 8.3.3 Media Access Control and Frame Format of Token Ring

As the name of the protocol suggests, the media access method used with Token Ring networks is called *token passing*. This is a deterministic access method that ensures no collisions will occur because only one station can transmit at any given time.

There are two types of frame for Token Ring LAN: token frame and data frame. A token frame is a short frame three octets in length, and can turn into a data frame when the token bit is set to 1, as shown in Fig. 8-6.

The token frame has a start delimiter (SD) and an end delimiter (ED), each with a length of one octet. The access control octet has four fields: priority, token indicator, monitor, and reserved bits. The priority field indicates the frame priority and a station can seize the token only if its own priority is equal to or higher than the token priority. The token indicator bit indicates whether the frame is a token or a data frame. The monitor field prevents any frame from circulating on the ring endlessly. The reserved bits field allows a station with higher priority to reserve the next token to be issued with the indicated priority.

A data frame is a superset of the token frame with additional fields such as destination and source addresses, data, and FCS fields, as shown in Fig. 8-6.

## 8.3.4 Token Ring LAN Operations

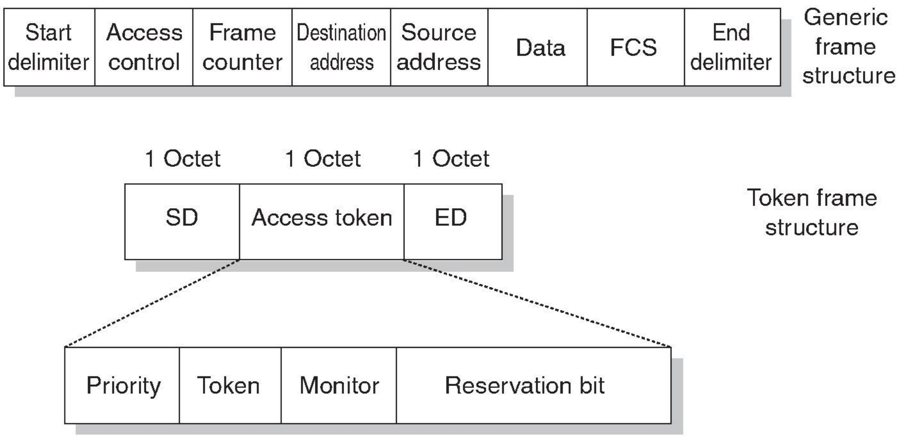
User data travels on the Token Ring network in one direction only, as in other ring topologies. The token frame is passed from node to node. Possession of the token gives a station permission to transmit data.

**8.3.4.1 Data Transmission Operations** When a station has data to send, it seizes the token first, then changes the token indicator bit to turn the token into the start of a data frame sequence. It then inserts user information and a destination address into the frame and sends it to the next station downstream on the ring.

Each station on the ring examines the data frame and passes it onto the next neighboring station if it is not the intended destination station. The destination station copies the frame for further processing, and sets a bit in the frame to acknowledge receipt of the frame.

The frame continues to circulate the ring until it reaches the sender. The sender removes the frame when it finds that the frame has been "seen" by the destination station. When the sender finishes sending the

**Figure 8-6** Token Ring frame structures.




last frame, it regenerates the token and puts it on the ring to allow other stations on the network to transmit data.

When a data frame is on the ring, no token frame is on the ring at the same time. This prevents two stations from transmitting data simultaneously so that no collisions occur.

**8.3.4.2 Priority** Token Ring LAN uses a priority system that allows the operator to assign high priority to some stations that can use the network more frequently than others. Briefly, the priority scheme works as follows: The token frame has two fields that control the priority: the priority field and the reservation field. Only those stations with a priority equal to or higher than the priority level contained in the token frame can seize the token. After the token is seized and changed to a data frame, those stations with a priority level higher than that of the transmitting station can reserve the token for the next round of token passing. When the next token frame is generated, it contains the higher-priority level of the reserving station. Once the reserving station finishes sending data, it is responsible for resetting the token frame's priority level to the original level in order to allow other stations a chance to transmit data.

**8.3.4.3 Ring Management** A station on a Token Ring LAN plays the role of either an active monitor or a standby monitor station. There is only one active monitor on a ring, and it is chosen during a process called the *claim token process*. The active monitor is responsible for maintaining the master clock, issuing a "neighbor notification," which is similar to a keep-alive message, detecting lost tokens and frames and purging the ring to get rid of endlessly circulating frames. Any station on the

ring can be the active monitor station if the current active monitor goes down, via the same claim token process.

# 8.4 FDDI

Fiber-distributed data interface (FDDI) LAN is another incarnation of Token Ring LAN, defined by ANSI (ANSI 1987, 1988) to fill two needs at the time the protocol was adopted. FDDI is intended to fill the need for a large amount of bandwidth on enterprise LANs and the need for reliable and fault-tolerant networks when enterprises start moving critical applications onto their networks. It was adopted by IEEE as IEEE 802.5 (IEEE 1998c), and by ISO. All the specifications are compatible and completely interoperable.

The FDDI standards define the physical layer and the data link layer of the LAN protocol stack. Specifically, they consist of four separate specifications covering the LAN physical layer protocol, PMD, MAC, and station management.

## 8.4.1 FDDI Basics

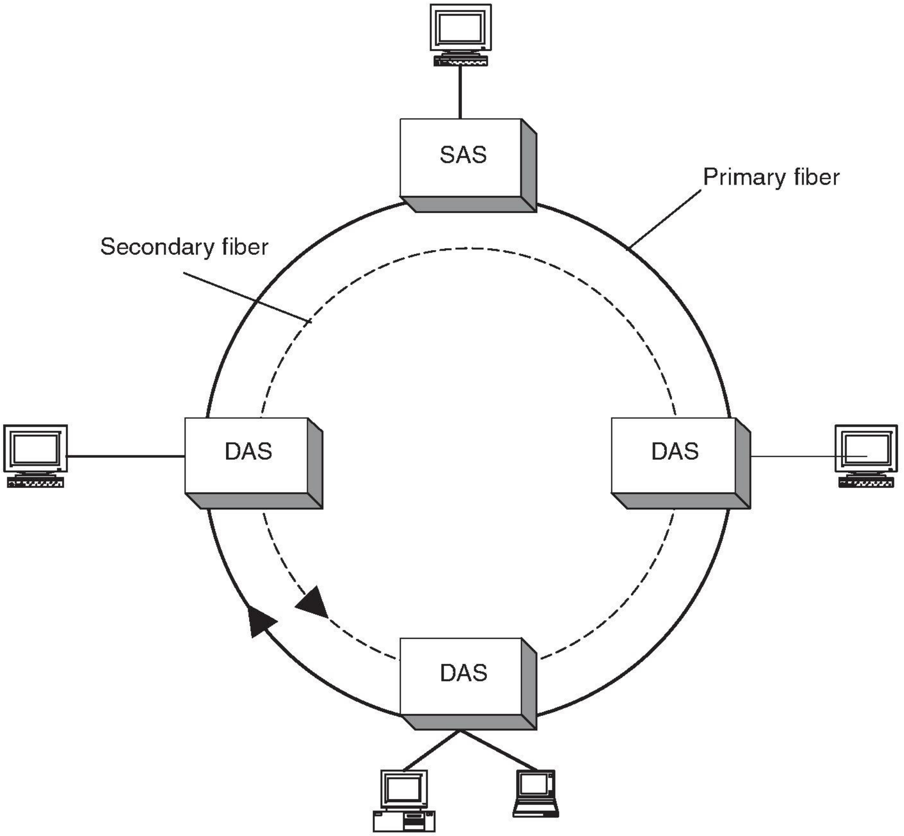
FDDI uses two types of optical fibers as primary transmission media: single-mode fiber, which is more expensive but has higher capacity, and multimode fiber, which is relatively inexpensive but has less capacity. The FDDI specification allows for 2 km between stations using multimode fiber and a longer distance with single-mode fiber, and supports a data rate of 100 Mbps.

The FDDI frame structure is very similar to that of the Token Ring frame structure described earlier, and it can be as large as 4500 octets. Like the Token Ring frame, the FDDI token frame is a subset of a general frame with three fields: a start delimiter, an end delimiter, and a frame control, which have identical fields to the token frame of Token Ring.

## 8.4.2 FDDI Configuration and Access Control

FDDI uses two counterrotating rings to enhance its fault tolerance capability: a primary ring and a secondary ring. As shown in Fig. 8-7, the secondary ring can be used to provide additional bandwidth or purely as a backup to the primary ring.

**Figure 8-7** Configuration and components of FDDI network.




In an FDDI LAN, there are two kinds of stations: the dual attachment station (DAS), which is connected to both rings, and the single attachment station (SAS), which is attached only to the primary ring. Another FDDI LAN device is the attachment concentrator, which allows multiple DASs or SASs to connect to either ring.

FDDI uses a media access control method that is different from that used by basic Token Ring. As discussed above, Basic Token Ring uses priority and reservation bits in the access control field of the token. In contrast, FDDI uses timed token rotation protocol, which operates as follows: For each rotation of the token, each station computes the time that has expired since it last received the token; this time is called the *token rotation time* (TRT). The TRT includes the time a station needs to transmit any of its waiting frames and the time all other stations in the ring need to transmit any of their waiting frames. TRT will be shorter if the system load is light and longer if the load is heavy. There is a predefined parameter called the *target token rotation time* (TTRT). Upon

receipt of a token, a station computes its TRT and the difference between the TTRT and the just computed TRT. The difference, known as the *token hold time* (THT), decides whether and how long the station can transmit the waiting frames. If the THT is positive, the station can spend up to the amount of time equal to the THT in transmitting data. If the THT is negative, the station cannot transmit any frame for this rotation of the token. This time token rotation protocol prevents a station from holding the token for an excessive amount of time and ensures that all stations have a fair chance to use it. This is the same mechanism the token bus protocol uses.

### 8.4.3 Station Management

There is one management station that acts as the manager on an FDDI ring, and each station has a station management agent. An agent station communicates with the management station to negotiate TTRT and reports the station status.

### 8.4.4 CDDI

A standard specification similar to FDDI for copper wire has emerged more recently, called the Copper Distributed Data Interface (CDDI) to be consistent with FDDI naming convention. CDDI is an implementation of the FDDI protocol on the copper medium and supports 100 Mbps over a 100-m distance from a desktop to a concentrator (ANSI 1995).

CDDI was defined by the ANSI X3T9.5 committee. It is officially named the Twisted-Pair Physical Medium-Dependent (TP-PMD) Standard to indicate that the focus of the specification is on the twisted pair physical medium, with rest of the protocol including the MAC algorithm and network configurations identical to that of FDDI.

## REVIEW QUESTIONS

1. What are the three media types for LAN? Describe the relationships between transmission distance and data rate.
2. The IEEE 802 LAN standards and protocols cover only the bottom two layers of the network reference model. Describe the responsibilities of each of the two bottom layers in the LAN context.

Local Area Networks



3. Describe the two media access control methods used for LANs and discuss the characteristics of each.
4. Describe the four LAN topologies and explain which ones are most commonly deployed.
5. Describe the operations of a LAN bridge and the differences between a LAN bridge and a LAN router.
6. Describe the responsibilities of the MAC and LLC sublayers in the LAN protocol stack.
7. Explain why Ethernet is a simple technology in terms of access control methods, network topology, and frame format.
8. Describe the differences between Fast Ethernet and the first generation of Ethernet in terms of transmission media, network topologies, and operation modes.
9. Describe the Token Ring LAN topology and how the token is passed around on a Token Ring LAN.
10. Describe the operations of the CSMA/CD access control method and compare it with the token-passing scheme.
11. Describe how the priority scheme in a Token Ring LAN allows some stations to transmit more data than other stations and how to prevent a frame from circulating the ring indefinitely.
12. Describe how the time token rotation protocol works as used in FDDI and token bus networks. Specifically, discuss how it prevents a station from holding onto the token for an excessive amount of time.
13. Briefly describe CDDI and compare it with FDDI.

# REFERENCES

ANSI. 1987. "FDDI—Token Ring Media Access Control (MAC)." ANSI X3.139. Web site: www.anci.org.

ANSI. 1988. "FDDI—Token Ring Physical Layer (PHY) Protocol." ANSI X3.148. Web site: www.anci.org.

ANSI. 1995. "FDDI—Token Ring Twisted Pair Physical Layer Medium Dependent (TP-PMD)." ANSI X3.263. Web site: www.anci.org.

ANSI. 1998. "Fibre Channel: Physical and Signaling Interface—3." ANSI INCITS 303. Web site: www.anci.org.

IEEE. 1990. "Token Passing Bus Access Method and Physical Layer Specifications." IEEE 802.4. Web site: www.ieee.org.

IEEE. 1994. "Distributed Queue Dual Bus (DQDB) Access Method and Physical Layer Specifications." IEEE 802.6. Web site: www.ieee.org.

IEEE. 1997. "IEEE Recommended Practices for Broadband Local Area Networks." IEEE 802.7. Web site: www.ieee.org.

IEEE. 1998a. "Demand-Priority Access Method, Physical Layer and Repeater Specifications." IEEE 802.12. Web site: www.ieee.org.

IEEE. 1998b. "Logical Link Control." IEEE 802.2. Web site: www.ieee.org.

IEEE. 1998c. "Token Ring Access Method and Physical Layer Specification." IEEE 802.5. Web site: www.ieee.org.

IEEE. 1999. "Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications." IEEE 802.11. Web site: www.ieee.org.

IEEE. 2001a. "Air Interface for Fixed Broadband Wireless Access Systems." IEEE 802.16. Web site: www.ieee.org.

IEEE. 2001b. "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications." 2001 edition. IEEE 802.3. Web site: www.ieee.org.

IEEE. 2001c. "Local and Metropolitan Area Networks: Overview and Architecture." IEEE 802. Web site: www.ieee.org.

IEEE. 2002a. "Media Access Control (MAC) Parameters, Physical Layer and Management Parameters for 10Gb/s Operation." IEEE 802.3ae. Web site: www.ieee.org.

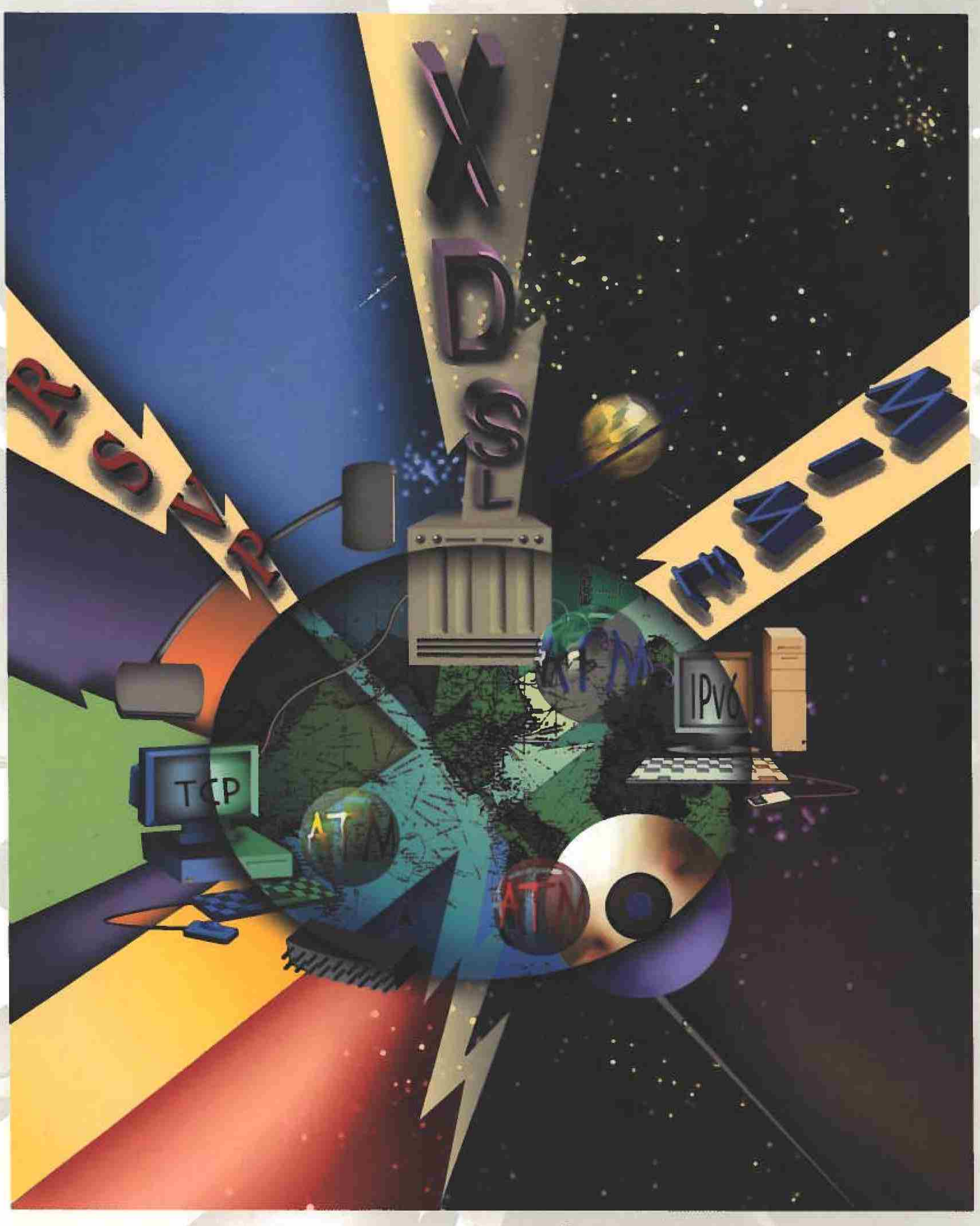
IEEE. 2002b. "Wireless Medium Access Control (MAC) and Physical Layer Specifications for Wireless Personal Area Networks (WPANs™)." IEEE 802.15. Web site: www.ieee.org.

Spurgen, C. 2000. *Ethernet: The Definitive Guide.* O'Reilly and Associations.

# Appendix H

# DATA & COMPUTER COMMUNICATIONS

SIXTH EDITION




WILLIAM STALLINGS

# DATA AND COMPUTER COMMUNICATIONS

SIXTH EDITION

William Stallings



Prentice Hall
Upper Saddle River, New Jersey 07458

**Library of Congress Cataloging-in-Publication Data**
Stallings, William.
Data and computer communications / William Stallings.—6th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-084370-9
1. Data transmission systems. 2. Computer networks. I. Title.
TK5105.S73 1999
004.6—dc20 99-24419
CIP

*Publisher*: Alan Apt
*Editor-in-chief*: Marcia Horton
*Project manager*: Ana Arias Terry
*Editorial/production supervision*: Rose Kernan
*Executive managing editor*: Vince O'Brien
*Copy editing*: Patricia Daly
*Art director and cover designer*: Heather Scott
*Assistant Vice-president of production*: David W. Riccardi
*Manufacturing buyer*: Pat Brown
*Editorial assistant*: Toni Holm



The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.



Printed in the United States of America

10 9 8 7 6 5 4 3 2

ISBN 0-13-084370-9



Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada, Inc., *Toronto*
Prentice-Hall Hispanoamericana, S. A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall (Singapore), Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

The discussion so far does not touch on other key aspects of data communications, including data link control techniques for controlling the flow of data and detecting and correcting errors, and multiplexing techniques for transmission efficiency. All of these topics are explored in Part Two.

## 1.3 DATA COMMUNICATION NETWORKING

It is often impractical for two communicating devices to be directly, point-to-point connected. This is so for one (or both) of the following contingencies:

- The devices are very far apart. It would be inordinately expensive, for example, to string a dedicated link between two devices thousands of kilometers apart.
- There is a set of devices, each of which may require a link to many of the others at various times. Examples are all of the telephones in the world and all of the terminals and computers owned by a single organization. Except for the case of a very few devices, it is impractical to provide a dedicated wire between each pair of devices.

The solution to this problem is to attach each device to a communication network. Figure 1.3 relates this discussion to the communications model of Figure 1.1a and also suggests the two major categories into which communications networks are traditionally classified: wide area networks (WANs) and local area networks (LANs). The distinction between the two, both in terms of technology and application, has become somewhat blurred in recent years, but it remains a useful way of organizing the discussion.

### Wide Area Networks

Wide area networks generally cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes (including the boundary nodes) are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination.

Traditionally, WANs have implemented using one of two technologies: circuit switching and packet switching. More recently, frame relay and ATM networks have assumed major roles.

#### Circuit Switching

In a circuit-switching network, a dedicated communications path is established between two stations through the nodes of the network. That path is a connected sequence of physical links between nodes. On each link, a logical channel is dedicated to the connection. Data generated by the source station are transmitted along the dedicated path as rapidly as possible. At each node, incoming data are routed

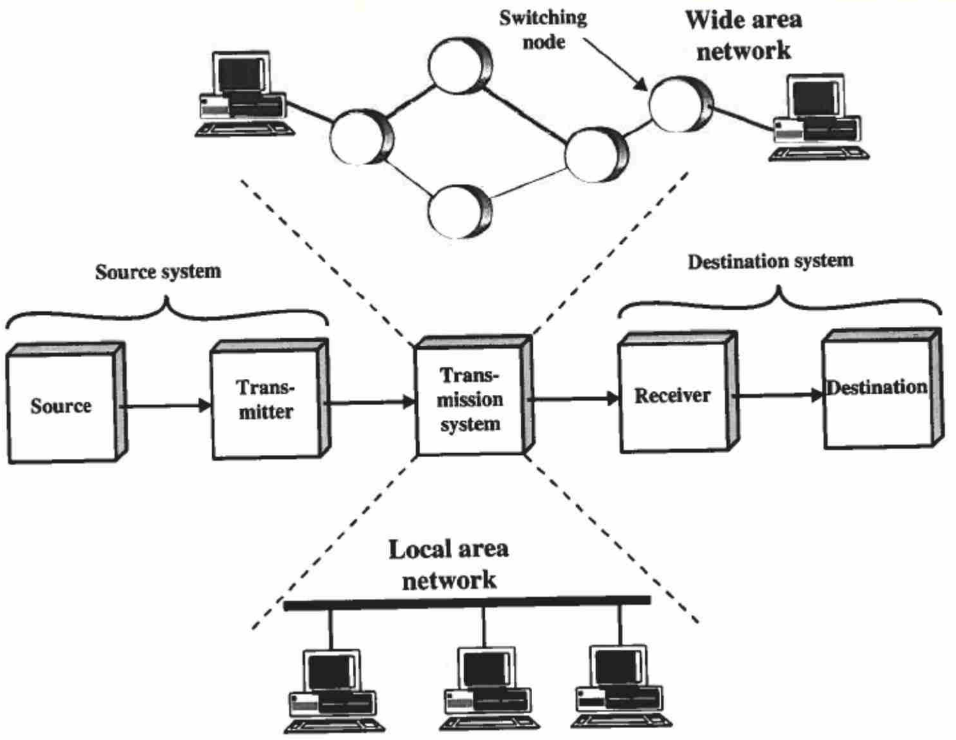



**Figure 1.3** Simplified Network Models

or switched to the appropriate outgoing channel without delay. The most common example of circuit switching is the telephone network.

### Packet Switching

A quite different approach is used in a packet-switching network. In this case, it is not necessary to dedicate transmission capacity along a path through the network. Rather, data are sent out in a sequence of small chunks, called packets. Each packet is passed through the network from node to node along some path leading from source to destination. At each node, the entire packet is received, stored briefly, and then transmitted to the next node. Packet-switching networks are commonly used for terminal-to-computer and computer-to-computer communications.

### Frame Relay

Packet switching was developed at a time when digital long-distance transmission facilities exhibited a relatively high error rate compared to today's facilities. As a result, there is a considerable amount of overhead built into packet-switching schemes to compensate for errors. The overhead includes additional bits added to each packet to introduce redundancy and additional processing at the end stations and the intermediate switching nodes to detect and recover from errors.

With modern high-speed telecommunications systems, this overhead is unnecessary and counterproductive. It is unnecessary because the rate of errors has been dramatically lowered and any remaining errors can easily be caught in the end systems by logic that operates above the level of the packet-switching logic. It is counterproductive because the overhead involved soaks up a significant fraction of the high capacity provided by the network.

Frame relay was developed to take advantage of these high data rates and low error rates. Whereas the original packet-switching networks were designed with a data rate to the end user of about 64 kbps, frame relay networks are designed to operate efficiently at user data rates of up to 2 Mbps. The key to achieving these high data rates is to strip out most of the overhead involved with error control.

### ATM

Asynchronous transfer mode (ATM), sometimes referred to as cell relay, is a culmination of all of the developments in circuit switching and packet switching over the past 25 years.

ATM can be viewed as an evolution from frame relay. The most obvious difference between frame relay and ATM is that frame relay uses variable-length packets, called frames, and ATM uses fixed-length packets, called cells. As with frame relay, ATM provides little overhead for error control, depending on the inherent reliability of the transmission system and on higher layers of logic in the end systems to catch and correct errors. By using a fixed packet length, the processing overhead is reduced even further for ATM compared to frame relay. The result is that ATM is designed to work in the range of 10s and 100s of Mbps, and in the Gbps range.

ATM can also be viewed as an evolution from circuit switching. With circuit switching, only fixed-data-rate circuits are available to the end system. ATM allows the definition of multiple virtual channels with data rates that are dynamically defined at the time the virtual channel is created. By using small, fixed-size cells, ATM is so efficient that it can offer a constant-data-rate channel even though it is using a packet switching technique. Thus, ATM extends circuit switching to allow multiple channels with the data rate on each channel dynamically set on demand.

### ISDN and Broadband ISDN

Merging and evolving communications and computing technologies, coupled with increasing demands for efficient and timely collection, processing, and dissemination of information, are leading to the development of integrated systems that transmit and process all types of data. A significant outgrowth of these trends is the integrated services digital network (ISDN).

The ISDN is designed to replace existing public telecommunications networks and deliver a wide variety of services. The ISDN is defined by the standardization of user interfaces and implemented as a set of digital switches and paths supporting a broad range of traffic types and providing value-added processing services. In practice, there are multiple networks, implemented within national boundaries, but from the user's point of view, there is a single, uniformly accessible, worldwide network.

Despite the fact that ISDN has yet to achieve the universal deployment hoped for, it is already in its second generation. The first generation, sometimes referred

to as **narrowband ISDN**, is based on the use of a 64-kbps channel as the basic unit of switching and has a circuit-switching orientation. The major technical contribution of the narrowband ISDN effort has been frame relay. The second generation, referred to as **broadband ISDN**, supports very high data rates (100s of Mbps) and has a packet-switching orientation. The major technical contribution of the broadband ISDN effort has been asynchronous transfer mode (ATM), also known as cell relay.

### Local Area Networks

As with WANs, a LAN is a communications network that interconnects a variety of devices and provides a means for information exchange among those devices. There are several key distinctions between LANs and WANs:

1. The scope of the LAN is small, typically a single building or a cluster of buildings. This difference in geographic scope leads to different technical solutions, as we shall see.
2. It is usually the case that the LAN is owned by the same organization that owns the attached devices. For WANs, this is less often the case, or at least a significant fraction of the network assets are not owned. This has two implications. First, care must be taken in the choice of LAN, because there may be a substantial capital investment (compared to dial-up or leased charges for WANs) for both purchase and maintenance. Second, the network management responsibility for a LAN falls solely on the user.
3. The internal data rates of LANs are typically much greater than those of WANs.

Traditionally, LANs make use of a broadcast network approach rather than a switching approach. With a broadcast communication network, there are no intermediate switching nodes. At each station, there is a transmitter/receiver that communicates over a medium shared by other stations. A transmission from any one station is broadcast to and received by all other stations. Data are usually transmitted in packets. Because the medium is shared, only one station at a time can transmit a packet.

More recently, examples of switched LANs, especially switched Ethernet LANs, have appeared. Two other prominent examples are ATM LANs, which simply use an ATM network in a local area, and Fibre Channel. We will examine these LANs, as well as broadcast LANs, in Part Four.

## 1.4 PROTOCOLS AND PROTOCOL ARCHITECTURE

When computers, terminals, and/or other data processing devices exchange data, the scope of concern is much broader than the concerns we have discussed in Sections 1.2 and 1.3. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed. Typical tasks to be performed:

# CHAPTER 13

# LAN TECHNOLOGY

- A LAN consists of a shared transmission medium and a set of hardware and software for interfacing devices to the medium and regulating the orderly access to the medium. The topologies that have been used for LANs are ring, bus, tree, and star. The bus and tree topologies are passive sections of cable to which stations are attached. A transmission of a frame by any one station can be heard by any other station. A ring LAN consists of a closed loop of repeaters that allow data to circulate around the ring. A repeater may also function as a device attachment point. Transmission is generally in the form of frames. A star LAN includes a central node to which stations are attached.
- The transmission media that are used for LANs are twisted pair, coaxial cable, optical fiber, and wireless. Both shielded and unshielded twisted pair are in use. Wireless transmission uses either infrared or microwave.
- A set of standards has been defined for LANs that specifies a range of data rates and encompasses all of the topologies and transmission media just mentioned. These standards, the IEEE 802 and fiber distributed data interface (FDDI) standards, are widely accepted, and most of the products on the market conform to one of these standards.
- In most cases, an organization will have multiple LANs that need to be interconnected. The simplest approach to meeting this requirement is the bridge.

We turn now to a discussion of local area networks (LANs). Whereas wide area networks may be public or private, LANs usually are owned by the organization that is using the network to interconnect equipment. LANs have much greater capacity than wide area networks, to carry what is generally a greater internal communications load.

A simple example of a LAN that highlights some of its characteristics is shown in Figure 1.3. All of the devices are attached to a shared transmission medium. A transmission from any one device can be received by all other devices attached to the same network. Traditional LANs have provided data rates in a range from about 1 to 20 Mbps. These data rates, though substantial, have become increasingly inadequate with the proliferation of devices, the growth in multimedia applications, and the increased use of the client/server architecture. Thus, recent efforts have focused on the development of high-speed LANs, with data rates of 100 Mbps to 1 Gbps.

This chapter begins our discussion of LANs[1] with a discussion of LAN application areas. This is followed by a description of the protocol architecture that is in common use for implementing LANs. This architecture is also the basis of standardization efforts. Our overview covers the physical, medium access control (MAC), and logical link control (LLC) levels.

[1]For the sake of brevity, the chapter often uses LAN when referring to LAN and metropolitan area network (MAN) concerns. The context should clarify when only LAN or both LAN and MAN is meant.

The key technology ingredients that determine the nature of a LAN or MAN are:

- Topology
- Transmission medium
- Medium access control technique

This chapter surveys the topologies and transmission media that are most commonly used for LANs. The issue of access control is briefly raised but is covered in more detail in Chapter 14. The concept of a bridge, which plays a critical role in extending LAN coverage, is discussed in Section 13.7.

## 13.1 LAN APPLICATIONS

The variety of applications for LANs is wide. This section provides a brief discussion of some of the most important general application areas for these networks.

### Personal Computer LANs

A common LAN configuration is one that supports personal computers. With the relatively low cost of such systems, individual managers within organizations often independently procure personal computers for departmental applications, such as spreadsheet and project management tools, and Internet access.

But a collection of department-level processors will not meet all of an organization's needs; central processing facilities are still required. Some programs, such as econometric forecasting models, are too big to run on a small computer. Corporate-wide data files, such as accounting and payroll, require a centralized facility but should be accessible to a number of users. In addition, there are other kinds of files that, although specialized, must be shared by a number of users. Further, there are sound reasons for connecting individual intelligent workstations not only to a central facility but to each other as well. Members of a project or organization team need to share work and information. By far the most efficient way to do so is digitally.

Certain expensive resources, such as a disk or a laser printer, can be shared by all users of the departmental LAN. In addition, the network can tie into larger corporate network facilities. For example, the corporation may have a building-wide LAN and a wide area private network. A communications server can provide controlled access to these resources.

LANs for the support of personal computers and workstations have become nearly universal in organizations of all sizes. Even those sites that still depend heavily on the mainframe have transferred much of the processing load to networks of personal computers. Perhaps the prime example of the way in which personal computers are being used is to implement client/server applications.

For personal computer networks, a key requirement is low cost. In particular, the cost of attachment to the network must be significantly less than the cost of the attached device. Thus, for the ordinary personal computer, an attachment cost in the hundreds of dollars is desirable. For more expensive, high-performance workstations, higher attachment costs can be tolerated. In any case, this suggests that the

# Appendix I

"I just don't see how anyone can run their network without it."
—Terè Parnell, Executive Technology Editor, *LAN Times*

LAN TIMES
OSBORNE

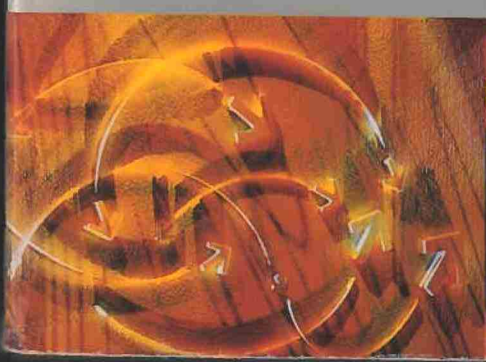
FEATURING FULLY SEARCHABLE, HYPERLINKED CD-ROM

# Encyclopedia of Networking

ELECTRONIC EDITION

TOM SHELDON Certified Network Engineer and Author of the Best-selling LAN Times Encyclopedia of Networking

- Thousands of Networking Terms and Concepts Thoroughly Explained
- Every Major Networking System Covered! UNIX, Windows NT, Novell NetWare, IBM, the Internet & More
- The Ideal Reference for Everyone Involved with Networks and the Internet





# Encyclopedia of Networking, Electronic Edition

Tom Sheldon

Osborne **McGraw-Hill**
Berkeley New York St. Louis San Francisco
Auckland Bogotà Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

**Osborne/McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations or book distributors outside the U.S.A., or to arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact Osborne/**McGraw-Hill** at the above address.

**Encyclopedia of Networking, Electronic Edition**



567890 DOC DOC 9019876543210

ISBN 0-07-882333-1 PPBK    ISBN 0-07-882350-1 HRDBK

**Publisher**
Brandon A. Nordin

**Editor-in-Chief**
Scott Rogers

**Acquisitions Editor**
Wendy Rinaldi

**Project Editor**
Emily Rader

**Editorial Assistant**
Ann Sellers

**Technical Editor**
Terè Parnell

**Copy Editor**
Dennis Weaver

**Proofreader**
Karen Mead

**Indexer**
Dan Logan

**Computer Designer**
Jani Beckwith

**Illustrators**
Sue Albert
Leslee Bassin
Arlette Crosland
Lance Ravella

**Cover Design**
Regan Honda

## L2F (Layer 2 Forwarding)

*See* Virtual Dial-up Services.

## L2TP (Layer 2 Tunneling Protocol)

*See* Virtual Dial-up Services.

## Label Switching

*See* IP over ATM; IP Switching; MPLS (Multiprotocol Label Switching); *and* Tag Switching.

## LAN (Local Area Network)

A LAN is a shared communication system to which many computers are attached. A LAN, as its name implies, is limited to a local area. This has to do more with the electrical characteristics of the medium than the fact that many early LANs were designed for departments, although the latter accurately describes a LAN as well.

LANs began to appear in the early 1970s. They grew from earlier point-to-point connections where a single wire connected two systems. Often the wire was quite long. Why not let multiple computers share this same cable? This required an arbitration mechanism to ensure that only one computer transmitted at once on the cable.

Arbitration methods are called *medium access controls*. Some methods have each workstation determine whether the cable is in use. Other methods use a central controller that gives each station access in turn. See "MAC (Medium Access Control) and Medium Access Control Methods" for more information on access methods.

LANs have different topologies, the most common being the *linear bus* and the *star configuration*. In the former, a cable snakes through a building from one workstation to another. In the star configuration, each workstation is connected to a central hub with its own cable. Each has its advantages and disadvantages. Interestingly, the most popular network, Ethernet, can take advantage of both topologies. Refer to "Topology" for more details.

A LAN is a connectionless networking scheme, meaning that once a workstation is ready to transmit and has access to the shared medium, it simply puts the packets on the network and hopes that the recipient receives them. There is no connection setup phase in this scheme. See "Connection-Oriented and Connectionless Services" for more details.

Data is packaged into *frames* for transmission on the LAN. At the hardware level, each frame is transmitted as a bit stream on the wire. Even though all the computers on the network listen to the transmission, only the designated recipient actually receives the frame. A frame is usually addressed for a single computer, although a *multicast address* can be used to transmit to all workstations on the LAN. Higher-layer protocols such as IP and IPX package data into datagrams. Datagrams are in turn

divided up and put into frames for transmission on a particular LAN. See "Datagrams and Datagram Services" and "Framing in Data Transmissions" for more details.

## LAN Distance and Size Limitations

One of the reasons why LANs are considered "local" is because there are practical limitations to the distance of a shared medium and the number of workstations you can connect to it. For example, if you tried to build a single LAN for an entire organization, there might be so many workstations attempting to access the cable at the same time that no real work would get done.

The electrical characteristics of the cable also dictate LAN limitations. Network designers must find a balance among the type of cable used, the transmission rates, signal loss over distance, and the signal emanation. All of these factors must stay within physical bounds and restrictions specified by various standards and government bodies. For example, coaxial cable allows higher transmission rates over longer distances, but twisted-pair wire is inexpensive, easy to install, and supports a hierarchical wiring scheme.

Delay is another factor. On Ethernet networks, workstations on either end of a long cable may not even detect that they are transmitting at the same time, thus causing a collision that results in corrupted data. You can use the following devices to extend a LAN or improve its performance:

- **Repeaters** Extends the limitations of Ethernet cable by boosting the signal. See "Repeater" for details.
- **Bridges** Provides repeater functions along with selective filtering of traffic to reduce congestion and contention. See "Bridges and Bridging" for details.
- **Switching** Provides an overall improvement in LAN throughput and design as described under "Switched Networks."
- **Routers** Provide a way to connect multiple LANs together to create internetworks. See "Internetworking" and "IP (Internet Protocol)" for more details.

**RELATED ENTRIES** Bridges and Bridging; Broadcast Networking; Connection Technologies; Data Communication Concepts; Datagram, and Datagram Services; Data Link Protocols; Ethernet; Framing in Data Transmissions; Internetworking; LAN Emulation; MAC (Medium Access Control); Medium Access Control Methods; Network Concepts; Network Design and Construction; Network Operating Systems; Packet; Protocol Concepts; Repeater; Switched Networks; Token Ring Network; Topology; VLAN (Virtual LAN); *and* Wireless Communications

## LAN Drivers

A LAN driver is a workstation or server software module that provides an interface between a NIC (network interface card) and the upper-layer protocol software running in the computer. The driver is designed for a specific NIC. Drivers are usually

Protocols specify a set of rules and procedures that define how communication takes place at different levels of operation. The lowest layers define physical connections, such as the cable type, access method, and topology, and how data is sent over the network. Further up are protocols that establish connections and maintain communication sessions between systems, and still further up are protocols that provide network interfacing for applications.

As mentioned, the OSI model has become the model to which all other network architectures and protocols are compared. The purpose of the OSI model is to coordinate communication standards between vendors. Refer to "OSI (Open Systems Interconnection) Model" for additional information.

**RELATED ENTRIES** Data Communication Concepts; Network Design and Construction; OSI (Open Systems Interconnection) Model; *and* Protocol Concepts

## Network Computer Devices

*See* NC (Network Computer) Devices.

## Network Concepts

A network is a communication system that allows users to access resources on other computers and exchange messages with other users. It allows users to share resources on their own systems with other network users and to access information on centrally located systems or systems that are located at remote offices. It may provide connections to the Internet or the networks of other organizations. Network connections allow users to operate from their home or on the road.

### The Scope of Networks

A network is a data communication system that links two or more computers and peripheral devices. As shown in Figure N-10, the network consists of a cable that attaches to NICs (network interface cards) in each of the devices. Figure N-11 illustrates the logical configuration of a network communication system. Users interact with network-enabled software applications to make a network request (such as to get a file or print on a network printer). The application communicates with the network software and the network software interacts with the network hardware. The network hardware is responsible for transmitting information to other devices attached to the network.

#### LAN (Local Area Network)

A LAN is a network that is located in a relatively small area, such as a department or building. Technically, a LAN consists of a shared medium to which workstations attach and communicate with one another using broadcast methods. With broadcasting, any device on the LAN can transmit a message that all other devices on the LAN can

N

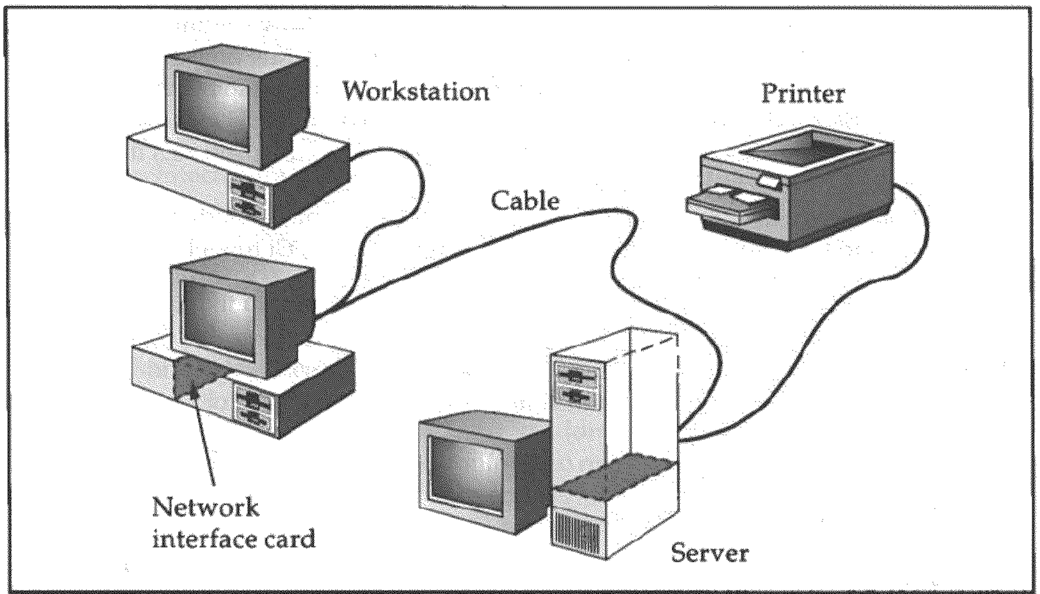



**Figure N-10.** *The components of a network*

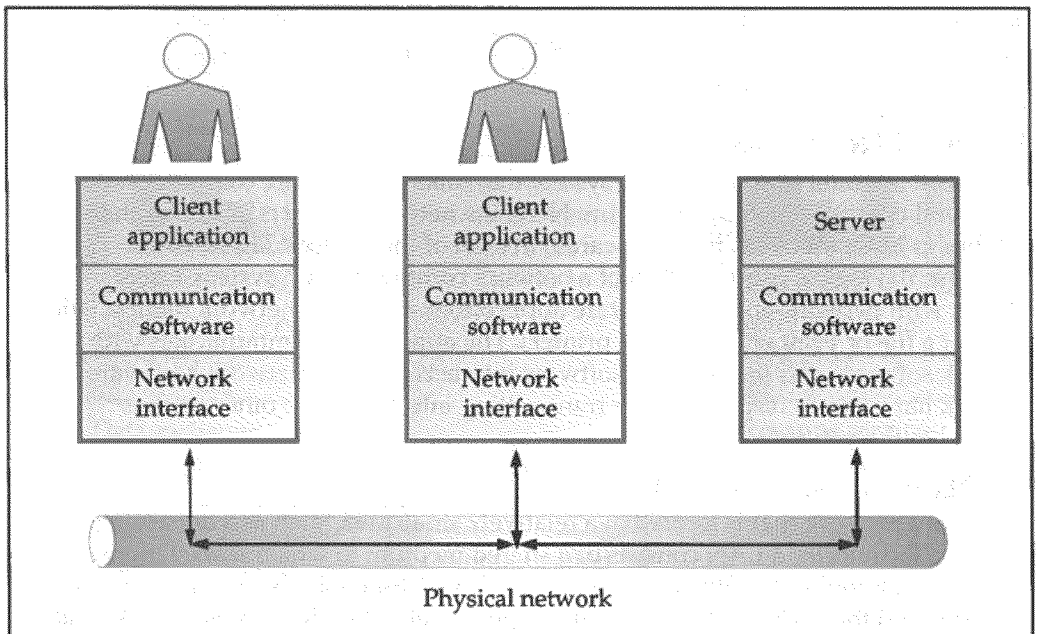



**Figure N-11.** *Logical configuration of a network*

listen to. The device to which the message is addressed actually receives the message. See "LAN (Local Area Network)" for more details.

Figure N-12 illustrates two ways to build a LAN. On the left, a relatively small LAN is built by running cable in a daisy-chain fashion from one department to the next. Each LAN segment is joined with a bridge. A bridge extends a LAN to create a much larger broadcast domain, but the bridge filters each individual segment's broadcasts by dropping frames that are not addressed to devices on connected segments. On the right, several LANs are interconnected at a centrally located hub device that handles the delivery of all inter-LAN traffic. See "Bridges and Bridging" and "Hubs/Concentrators/MAUs" for more information.

The model on the right in Figure N-13 implements a structured wiring system that is hierarchical in nature. Cables branch from a central internetwork hub to departmental hubs. This system of interconnecting cables and hub is often referred to as the backbone network. See "Backbone Networks" for more information.

The two most popular LAN technologies are Ethernet and token ring. See "Ethernet" and "Token Ring Network" for more details.

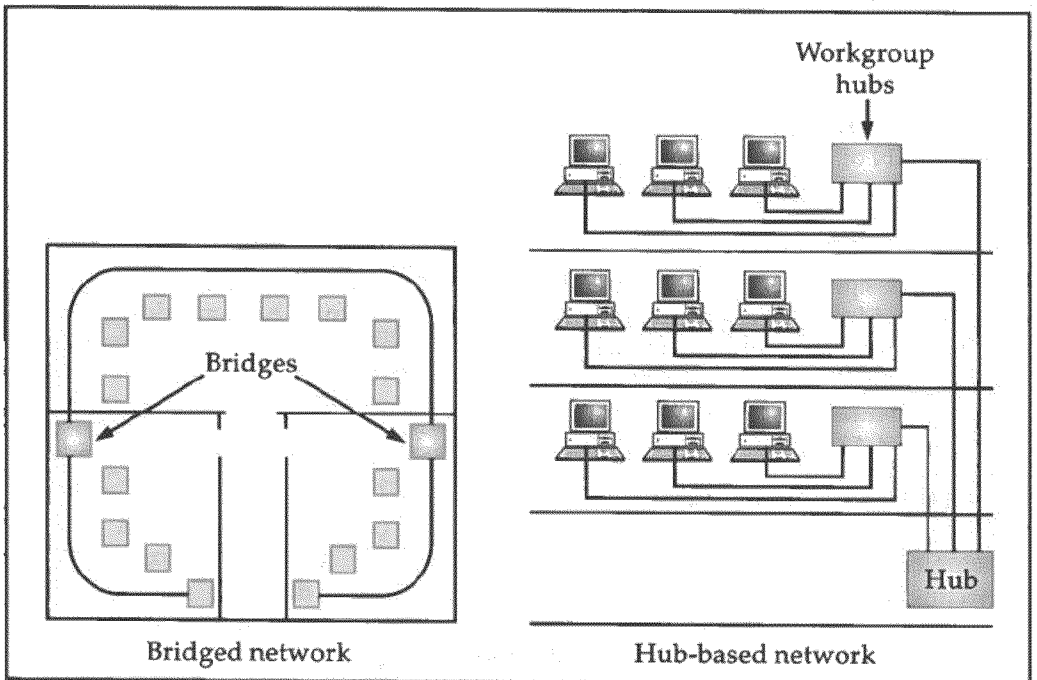



**Figure N-12.** *A linear network design (left) and a structured network design (right)*

N

# Appendix J

Google

google dictionary

All Books News Images Shopping More Settings Tools

About 572,000,000 results (0.70 seconds)

## Dictionary

local area network

# lo·cal ar·e·a net·work

/ˈ ˌlōkəl ˈ ˌerēə ˈnetˌwərk/

*noun*

noun: **local area network**; plural noun: **local area networks**; noun: **LAN**; plural noun: **LANs**

a computer network that links devices within a building or group of adjacent buildings.

# Appendix K

Joseph Migga Kizza

# A Guide to Computer Network Security

Springer

Joseph Migga Kizza, PhD
University of Tennessee-Chattanooga
Department of Computer Science
615 McCallie Ave.
Chattanooga TN 37403
326 Grote Hall
USA
joseph-kizza@utc.edu

Series Editor
Professor A.J. Sammes, BSc, MPhil, PhD, FBCS, CEng
CISM Group, Cranfield University,
RMCS, Shrivenham, Swindon SN6 8LA,UK

CCN Series ISSN 1617-7975
ISBN 978-1-84800-916-5 e-ISBN 978-1-84800-917-2
DOI 10.1007/978-1-84800-917-2

Library of Congress Control Number: 2008942999



Printed on acid-free paper

springer.com

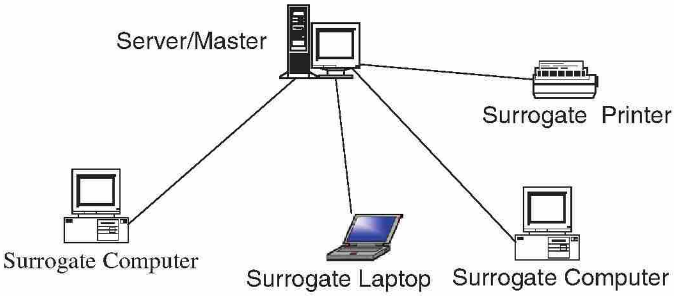



**Fig. 1.2** A Centralized network model

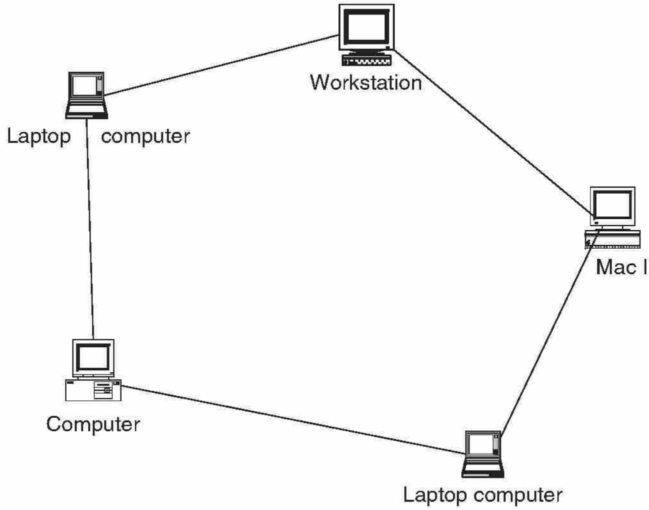



**Fig. 1.3** A Distributed network model

## 1.3 Computer Network Types

Computer networks come in different sizes. Each network is a cluster of network elements and their resources. The size of the cluster determines the network type. There are, in general, two main network types: the local area network (LAN) and wide area network (WAN).

### *1.3.1 Local Area Networks (LANs)*

A computer network with two or more computers or clusters of network and their resources connected by a communication medium sharing communication protocols and confined in a small geographical area, such as a building floor, a building,

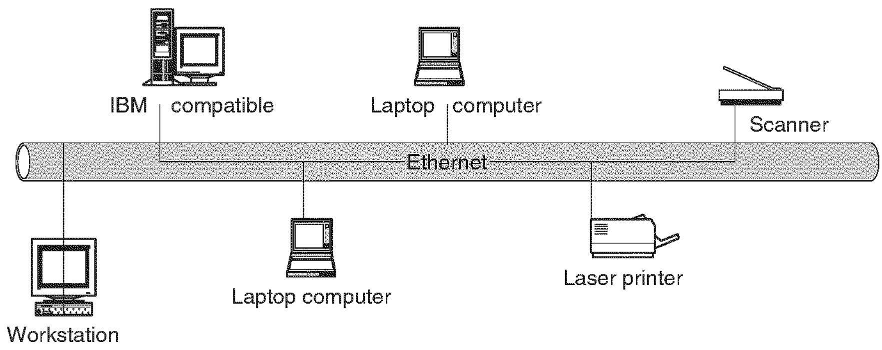



**Fig. 1.4** A LAN Network

or a few adjacent buildings, is called a local area network (LAN). The advantage of a LAN is that all network elements are close together so the communication links maintain a higher speed of data movement. Also, because of the proximity of the communicating elements, high-cost and high quality communicating elements can be used to deliver better service and high reliability. Figure 1.4 shows a LAN network.

### *1.3.2 Wide Area Networks (WANs)*

A wide area network (WAN), on the other hand, is a network made up of one or more clusters of network elements and their resources but instead of being confined to a small area, the elements of the clusters or the clusters themselves are scattered over a wide geographical area as in a region of a country or across the whole country, several countries, or the entire globe like the Internet for example. Some advantages of a WAN include distributing services to a wider community and availability of a wide array of both hardware and software resources that may not be available in a LAN. However, because of the large geographical areas covered by WANs, communication media are slow and often unreliable. Figure 1.5 shows a WAN network.

### *1.3.3 Metropolitan Area Networks (MANs)*

Between the LAN and WAN, there is also a middle network called the metropolitan area network (MAN) because it covers a slightly wider area than the LAN but not so wide as to be considered a WAN. Civic networks that cover a city or part of a city are a good example of a MAN. MANs are rarely talked about because they are quiet often overshadowed by cousin LAN to the left and cousin WAN to the right.

# Appendix L

FACTS ON FILE SCIENCE LIBRARY

# ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

HARRY HENDERSON

REVISED EDITION

# ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

REVISED EDITION

HARRY HENDERSON

Facts On File
An imprint of Infobase Publishing

*In memory of my brother,*
*Bruce Henderson,*
*who gave me my first opportunity to explore*
*personal computing almost 30 years ago.*

**ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY, Revised Edition**



Facts On File, Inc.
An imprint of Infobase Publishing
132 West 31st Street
New York NY 10001

**Library of Congress Cataloging-in-Publication Data**

Henderson, Harry, 1951–
Encyclopedia of computer science and technology / Harry Henderson.—Rev. ed.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-8160-6382-6
ISBN-10: 0-8160-6382-6
1. Computer science—Encyclopedias. 2. Computers—Encyclopedias. I. Title.
QA76.15.H43 2008
004.03—dc22 2008029156



Text design by Erika K. Arroyo
Cover design by Salvatore Luongo
Illustrations by Sholto Ainslie
Photo research by Tobi Zausner, Ph.D.

Printed in the United States of America

VB Hermitage 10 9 8 7 6 5 4 3 2 1

This book is printed on acid-free paper and contains
30 percent postconsumer recycled content.

Petzold, Charles. *Programming Windows: the Definitive Guide to the Win32 API*. 5th ed. Redmond, Wash.: Microsoft Press, 1999.
"Windows API Guide." Available online. URL: http://www.vbapi.com/. Accessed April 12, 2007.

## application service provider (ASP)

Traditionally, software applications such as office suites are sold as packages that are installed and reside on the user's computer. Starting in the mid-1990s, however, the idea of offering users access to software from a central repository attracted considerable interest. An application service provider (ASP) essentially rents access to software.

Renting software rather than purchasing it outright has several advantages. Since the software resides on the provider's server, there is no need to update numerous desktop installations every time a new version of the software (or a "patch" to fix some problem) is released. The need to ship physical CDs or DVDs is also eliminated, as is the risk of software piracy (unauthorized copying). Users may be able to more efficiently budget their software expenses, since they will not have to come up with large periodic expenses for upgrades. The software provider, in turn, also receives a steady income stream rather than "surges" around the time of each new software release.

For traditional software manufacturers, the main concern is determining whether the revenue obtained by providing its software as a service (directly or through a third party) is greater than what would have been obtained by selling the software to the same market. (It is also possible to take a hybrid approach, where software is still sold, but users are offered additional features online. Microsoft has experimented with this approach with its Microsoft Office Live and other products.)

Renting software also has potential disadvantages. The user is dependent on the reliability of the provider's servers and networking facilities. If the provider's service is down, then the user's work flow and even access to critical data may be interrupted. Further, sensitive data that resides on a provider's system may be at risk from hackers or industrial spies. Finally, the user may not have as much control over the deployment and integration of software as would be provided by outright purchase.

The ASP market was a hot topic in the late 1990s, and some pundits predicted that the ASP model would eventually supplant the traditional retail channel for mainstream software. This did not happen, and more than a thousand ASPs were among the casualties of the "dot-com crash" of the early 2000s. However, ASP activity has been steadier if less spectacular in niche markets, where it offers more economical access to expensive specialized software for applications such as customer relationship management, supply chain management, and e-commerce related services—for example, Salesforce.com. The growing importance of such "software as a service" business models can be seen in recent offerings from traditional software companies such as SAS. By 2004, worldwide spending for "on demand" software had exceeded $4 billion, and Gartner Research has predicted that in the second half of the decade about a third of all software will be obtained as a service rather than purchased.

### *Web-Based Applications and Free Software*

By that time a new type of application service provider had become increasingly important. Rather than seeking to gain revenue by selling online access to software, this new kind of ASP provides the software for free. A striking example is Google Pack, a free software suite offered by the search giant (see GOOGLE). Google Pack includes a variety of applications, including a photo organizer and search and mapping tools developed by Google, as well as third-party programs such as the Mozilla Firefox Web browser, RealPlayer media player, the Skype Internet phone service (see VOIP), and antivirus and antispyware programs. The software is integrated into the user's Windows desktop, providing fast index and retrieval of files from the hard drive. (Critics have raised concerns about the potential violation of privacy or misuse of data, especially with regard to a "share across computers" feature that stores data about user files on Google's servers.) America Online has also begun to provide free access to software that was formerly available only to paid subscribers.

This use of free software as a way to attract users to advertising-based sites and services could pose a major threat to companies such as Microsoft that rely on software as their main source of revenue. In 2006 Google unveiled a Google Docs & Spreadsheets, a program that allows users to create and share word-processing documents and spreadsheets over the Web. Such offerings, together with free open-source software such as Open Office.org, may force traditional software companies to find a new model for their own offerings.

Microsoft in turn has launched Office Live, a service designed to provide small offices with a Web presence and productivity tools. The free "basic" level of the service is advertising supported, and expanded versions are available for a modest monthly fee. The program also has features that are integrated with Office 2007, thus suggesting an attempt to use free or low-cost online services to add value to the existing stand-alone product line.

By 2008 the term *cloud computing* had become a popular way to describe software provided from a central Internet site that could be accessed by the user through any form of computer and connection. An advantage touted for this approach is that the user need not be concerned with where data is stored or the need to make backups, which are handled seamlessly.

### Further Reading

Chen, Anne. "Office Live Makes Online Presence Known." *eWeek*, November 2, 2006. Available online. URL: http://www.eweek.com/article2/0,1759,2050580,00.asp. Accessed May 22, 2007.
Focacci, Luisa, Robert J. Mockler, and Marc E. Gartenfeld. *Application Service Providers in Business*. New York: Haworth, 2005.
Garretson, Rob. "The ASP Reincarnation: The Application Service Provider Name Dies Out, but the Concept Lives on among Second-Generation Companies Offering Software as a service." *Network World*, August 29, 2005. Available online.

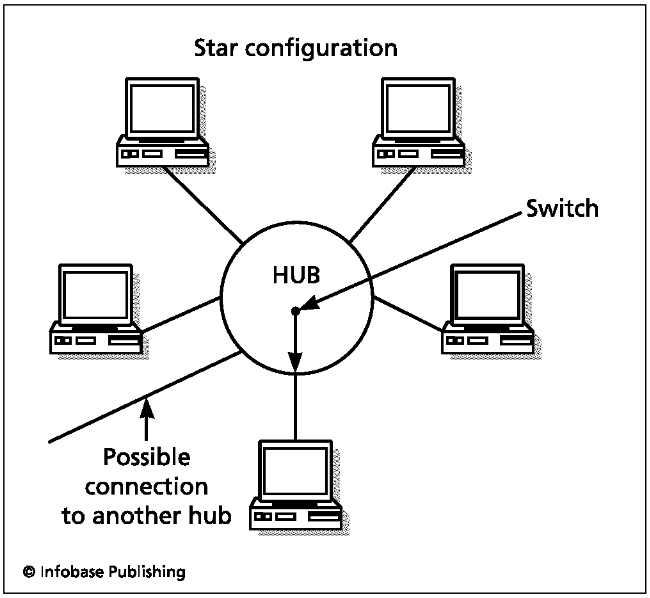
edu/mc/LispNotes/RecursiveListProcessingAlgorithmsIn-Lisp.pdf. Accessed August 13, 2007.
"Linked List Basics [In C/C++]" Available online. URL: http://cslibrary.stanford.edu/103/LinkedListBasics.pdf. Accessed August 13, 2007.
Seibel, Peter. "They Called It LISP for a Reason: List Processing." Available online. URL: http://www.gigamonkeys.com/book/they-called-it-lisp-for-a-reason-list-processing.html. Accessed August 13, 2007.

## local area network (LAN)

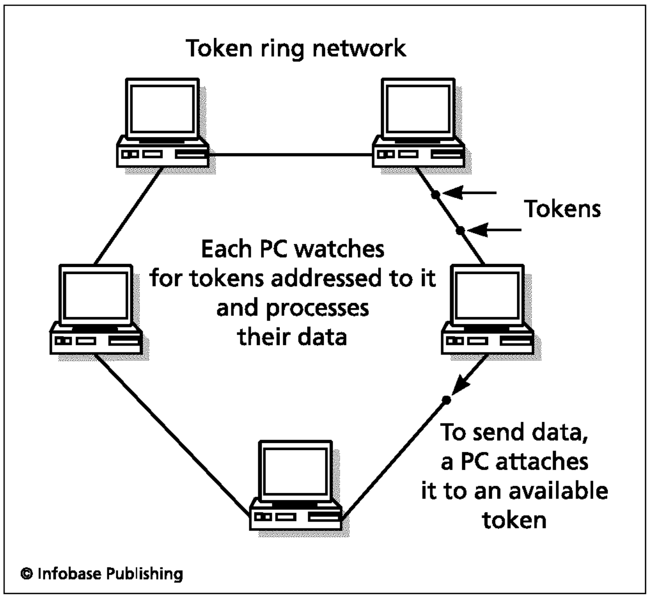
Starting in the 1980s, many organizations sought to connect their employees' desktop computers so they could share central databases, share or back up files, communicate via e-mail, and collaborate on projects. A system that links computers within a single office or home, or a larger area such as a building or campus, is called a local area network (LAN). (Larger networks linking branches of an organization throughout the country or world are called wide area networks, or WANs. See NETWORK.)

### HARDWARE ARCHITECTURE

There are two basic ways to connect computers in a LAN. The first, called Ethernet, was developed by a project at the Xerox Palo Alto Research Center (PARC) led by Robert Metcalfe. Ethernet uses a single cable line called a *bus* to which all participating computers are connected. Each data packet is received by all computers, but processed only by the one it is addressed to. Before sending a packet, a computer first checks to make sure the line is free. Sometimes, due to the time delay before a packet is received by all computers, another computer may think the line is free and start transmitting. The resulting *collision* is resolved by having both computers stop and wait varying times before resending.




*The Star network configuration uses a central hub to which each PC is attached. To extend the network (such as into other offices), the hubs can be connected to one another so they function as switches. When a token arrives that is addressed to one of its PCs, the hub will route it to the appropriate machine.*




*A Token Ring network connects the machines in a "chain" around which messages called tokens travel. Any PC can "grab" a passing token and attach data and the address of another PC to it. Each PC in turn watches for tokens that are addressed to it.*

Because connecting all computers to a single bus line is impractical in larger installations, Ethernet networks are frequently extended to multiple offices by connecting a bus in each office to a switch, creating a subnetwork or segment (this is sometimes called a *star topology*). The switches are then connected to a main bus. Packets are first routed to the switch for the segment containing the destination computer. The switch then dispatches the packet to the destination computer. Another advantage of this *switched Ethernet* system is that more-expensive, high-bandwidth cable can be used to connect the switches to move the packets more quickly over greater distances, while less-expensive cabling can be used to connect each computer to its local switch.

An alternative way to arrange a LAN is called *token ring*. Instead of the computers being connect to a bus that ends in a terminator, they are connected in a circle where the last computer is connected to the first. Interference is prevented by using a special packet called the token. Like the use of a "talking stick" in a tribal council, only the computer holding the token can transmit at a given time. After transmitting, the computer puts the token back into circulation so it can be grabbed by the next computer that wants to send data.

### LAN SOFTWARE

Naturally there must be software to manage the transmission and reception of data packets. The structure of a packet (sometimes called a *frame*) has been standardized with a preamble, source and destination addresses, the data itself,

# Appendix M

# NEWTON's TELECOM DICTIONARY

**STAY INFORMED**

To be alerted by email to updates and corrections go to www.cmpbooks.com/newton

CMPBooks
San Francisco

**NEWTON's TELECOM DICTIONARY**


email: Harry@HarryNewton.com
personal web site: www.HarryNewton.com
business web site: www.TechnologyInvestor.com



Published by CMP Books
An imprint of CMP Media LLC
Main office: CMP Books, 600 Harrison St., San Francisco, CA 94107 USA
Phone: 415-947-6615; Fax: 415-947-6015
Sales office: CMP Books, 12 West 21 Street, New York, NY 10010
Phone 917-305-3333; Fax 212-206-0387
www.cmpbooks.com
Email: books@cmp.com

CMP
United Business Media

For individual orders, and for information on special discounts for quantity orders, please contact:
CMP Books Distribution Center, 6600 Silacci Way, Gilroy, CA 95020
Tel: 1-800-500-6875 or 408-848-3854; Fax: 408-848-5784
Email: cmp@rushorder.com; Web: www.cmpbooks.com

Distributed to the book trade in the U.S. by:
Publishers Group West, 1700 Fourth Street, Berkeley, California 94710

Distributed in Canada by:
Jaguar Book Group, 100 Armstrong Avenue, Georgetown, Ontario M6K 3E7 Canada

Printed in the United States of America

This book is also sold through www.Amazon.com, www.Fatbrain.com and www.BarnesAndNoble.com

Distributed to the book trade in the U.S. and Canada by Publishers Group West
1700 Fourth St., Berkeley, CA 94710
Fax: 408-848-5784
cmp@rushorder.com

ISBN Number 1-57820-307-4

March 2003

Nineteenth Edition

Matt Kelsey, Publisher
Ray Horak, Senior Contributing Editor
Saul Roldan, Cover Artist
Lisa Giaquinto, Project Manager
Brad Greene, Text Layout

**Loading Plan** A telephone company term. A Loading Plan is a systematic scheme for fully utilizing all existing capacity in a given switching entity; Utilizing and coordinating the capabilities and capacity limitations of various entities in a multi-entity wire center and maintaining objective service levels at all times. A Loading Plan is the basis for achieving and retaining good Load Balance.

**LOC** An ATM term. Loss of Cell Delineation: A condition at the receiver or a maintenance signal transmitted in the PHY overhead indicating that the receiving equipment has lost cell delineation. Used to monitor the performance of the PHY layer.

**Local** Pertaining to a system or device that resides within a subject device's switching domain.

**Local Access** The connection between a customer's premises and a point of presence of the Exchange Carrier.

**Local Access and Transport Area** LATA. The MFJ (Modified Final Judgement), which broke up the Bell System, also defined 196 distinct geographical areas known as LATAs. The LATA boundaries generally were drawn in consideration of SMSAs (Standard Metropolitan Statistical Areas), which were defined by the Census Bureau to identify "communities of interest" in economic terms. Generally speaking, the LATA boundaries also were coterminous with state lines and existing area code boundaries, and generally included the territory served by only a single RBOC. The basic purpose of the LATA concept was to delineate the serving areas reserved for LEC (Local Exchange Carrier) activity. In other words, IntraLATA traffic (i.e., local and local long distance) became the sole right and responsibility of the LECs. InterLATA traffic, on the other hand, became the sole right and responsibility of the IXCs. Over time, a number of state PUCs allowed the IXCs to compete for IntraLATA long distance; they also allowed CAPs (Competitive Access Providers) to provided limited local service in competition with the LECs. The Telecommunications Act of 1996 (The Act) opened the floodgates for competition with the LATA boundaries. The Act also allows the RBOCs to provide InterLATA service outside the states in which they provide local service. Additionally, The Act contains provisions for the RBOCs to offer InterLATA service within the state in which they provide local service, once they have satisfied a 14-point checklist, the most significant conditions of which relate to significant, demonstrated levels of competition within their respective local exchange serving areas. California is divided into 10 LATAs. Sparsely populated states such as South Dakota comprise only a single LATA.

**Local Airtime Detail** This cellular telephone carrier option (which means it costs money) provides a line-itemized, detailed billing of all calls, including call attempts and incoming calls to the mobile. What you get for free is generally a non-detailed, total summary of all calls.

**Local Area And Transport Area** See LATA.

**Local Area Data Transport** LADT. A service of your local phone company which provides you, the user, with synchronous data communications.

**Local Area Network** LAN. A short distance data communications network (typically within a building or campus) used to link computers and peripheral devices (such as printers, CD-ROMs, modems) under some form of standard control. Older data communications networks used dumb terminals (devices with no computing power) to talk to distant computers. But the economics of computing changed with the invention of the personal computer which had "intelligence" and which was cheap. LANs were invented as an afterthought — after PCs — and were originally designed to let cheap PCs share peripherals — like laser printers — which were too expensive to dedicate to individual PCs. And as time went on, what LANs were used for got broader and broader. Today, LANs have four main advantages: 1. Anyone on the LAN can use any of the peripheral devices connected to the LAN. 2. Anyone on the LAN can access databases and programs running on client servers (super powerful PCs) attached to the LAN; and 3. Anyone on the LAN can send messages to and work jointly with others on the LAN. 4. While a LAN does not use common carrier circuits, it may have gateways and/or bridges to public telecommunications networks. See LAN Manager, Token Ring and Ethernet.

**Local Area Signaling Services** LASS is a group of central office features provided now by virtually all central office switch makers that uses existing customer lines to provide some extra features to the end user (typically a business user). They are based on delivery of calling party number via the local signaling network. LASS can be implemented on a standalone single central office basis for intra office calls or on a multiple central office grouping in a LATA (what the local phone companies are allowed to serve) for interoffice calls. Local CCS7 (Common Channel Signaling Seven) is required for all configurations. The following features typically make up LASS:

Automatic Callback: Lets the customer automatically call the last incoming call directory number associated with the customer's phone when both phones become idle. This feature gives the customer the ability to camp-on to a line.

Automatic Recall: Lets the customer automatically call the last outgoing call currently associated with the customer's station when both stations become idle. This feature gives the customer the ability to camp-on to a line.

Customer-Originated Trace: Lets the terminating party request an automatic trace of the last call received. The trace includes the calling line directory number and time and date of the call. This information is transmitted via an AM IOP channel to a designated agency, such as the telephone company or law enforcement agency.

Individual Calling Line Identification: Consists of two distinct features: 1. Calling Number Delivery which transmits data on an incoming call to the terminating phone. 1. Directory Number Privacy which prevents delivery of the directory number to the terminating phone.

Also, LASS has some selective features:

Selective Call Acceptance: Allows users to restrict which incoming voice calls can terminate, based on the identity attribute of the calling party. Only calls from parties identified on a screening lists are allowed to terminate. Calls from parties not specified on a screening list are rerouted to an appropriate announcement or forwarded to an alternate directory number.

Selective Call Forwarding: Allows a customer to pre-select which calls are forwarded based on the identity attribute of the calling party.

Selective Call Rejection: Allows a customer to reject incoming voice calls from identity attributes which are on the customer's rejection list. Call attempts from parties specified on the rejection list are prevented from terminating to the customer and are routed to an announcement which informs the caller that his/her call is not presently being accepted by the called party.

Selective Distinctive Alert: Allows a customer to pre-select which voice calls are to be provided distinctive alerting treatment based on the identify attributes of the calling party.

Users can, at their convenience, activate or modify any of these features by sending commands to the central switch from their existing touchtone telephones.

**Local Attack** A network security term. An attack that targets the machine on which the attacker is interactively logged on.

**Local Automatic Message Accounting** LAMA. A combination of automatic message accounting equipment and automatic number identification equipment in your telephone company's central office and used by them to bill your local phone calls.

**Local Battery** Having "local battery" means the telecom equipment — the telephone, the PBX, the key system, etc. — has its own source of power and does not draw from the power coming down the phone line. The term came from telegraphy and was used to distinguish the battery which provided power to the telegraphic station as against the power that went to drive the line and the signal traveling down it. See Battery.

**Local Bridge** A bridge between two or more similar networks on a local site (within same building).

**Local Bus** A microprocessor inside a PC must communicate with certain integral devices, including memory, video controllers, hard disks. This is typically called an internal bus. That is to distinguish it from the "external" bus, such as the AT, ISA, EISA, MCA buses, which define the communications between the motherboard and the various peripheral devices, such as the I/O cards like those handling modems and LAN connections. As microprocessors have gotten faster, so they have begun to outpace the speed of their computer's internal bus, which has tended to narrow the stream of data in and out of the CPU, slowing the computer. A Local Bus is a new type of internal bus. It is a faster bus. The idea is to get a broader path between your critical components — memory, video and disk controller — and your microprocessor. The idea is to get the data in and out of the microprocessor at the same speed as the microprocessor's system clock. Local Bus is an emerging standard. See also EISA, PCI and VESA.

**Local Call** Any call within the local service area of the calling phone. Individual local calls may or may not cost money. In many parts of the US, the phone company bills its local service as a "flat" monthly fee. This means you can make as many local calls per month as you wish and not pay extra. Increasingly this luxury is dying and local calls are costing money.

**Local Call Accounting** Computes the dollar amount for local calls based on the total message units stored for each phone.

**Local Call Billing** Computes the dollar amount for local calls placed by guests based on total message units.

L

# Appendix N

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**(Attorney Docket No. 11-1001-CON0115 (MBHB 14-1795-US3))**

| | | |
|---|---|---|
| In the Application of: | ) | |
| Arthur Coburn IV | ) | Examiner: Oschta Montoya |
| | ) | |
| Application No.: 14/628,952 | ) | Group Art Unit: 2421 |
| | ) | |
| Filing Date: Feb. 23, 2015 | ) | Confirmation No.: 6897 |
| | ) | |
| For: Networked Music Playback | ) | |
| | ) | |

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313

**RESPONSE TO NON-FINAL OFFICE ACTION MAILED JULY 25, 2016**

Responsive to the Non-Final Office Action mailed July 25, 2016, Applicant respectfully requests reconsideration of the application in view of the following remarks. Applicant generally authorizes the Office to charge any underpayment or credit any overpayment to Deposit Account No. 13-2490 and to treat this or any subsequent communication that requires an extension of time as incorporating a request for such an extension.

**Amendments to the Claims** begin on page 2.

**Remarks** begin on page 14.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

## AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method comprising:

causing, via a control device, a graphical interface to display a control interface including one or more transport controls to control playback by the control device;

after connecting to a local area network via a network interface, identifying, via the control device, playback devices connected to the local area network;

causing, via the control device, the graphical interface to display a selectable option for transferring playback from the control device;

detecting, via the control device, a set of inputs to transfer playback from the control device to a particular playback device, wherein ~~detecting~~ the set of inputs comprises: (i) a selection of the selectable option for transferring playback from the control device and (ii) a selection of the particular playback device from the identified playback devices connected to the local area network;

~~detecting, via the control device, a first input comprising an identification of the playback device;~~

~~detecting, via the control device, a second input comprising an identification of an item, wherein multimedia content associated with the item is retrievable from a content provider;~~

~~detecting, via the control device, a trigger, wherein detecting the trigger comprises detecting one or more third inputs that are not the first input or the second input; and~~

after detecting the set of inputs to transfer playback from the control device to the particular playback device, causing playback to be transferred from the control device to the particular playback device, wherein transferring playback from the control device to the particular playback device comprises:

(a) causing one or more first cloud servers to add multimedia content to a local playback queue on the particular playback device;

(b) causing playback at the control device to be stopped; and

(c) modifying the one or more transport controls of the control interface to control playback by the playback device; and



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

causing the particular playback device to play back the multimedia content, wherein the particular playback device playing back the multimedia content comprises the particular playback device retrieving the multimedia content from one or more second cloud servers of a streaming content service and playing back the retrieved multimedia content

~~sending, via a network interface, information regarding the multimedia content from the control device to the playback device, wherein the information comprises an identification of the multimedia content for playback by the playback device, and wherein the information causes the playback device to (a) retrieve, independent of the control device, the multimedia content from the content provider and (b) initiate playback of the retrieved multimedia content~~.

2. (Currently amended) The method of claim 1, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone of a media playback system that includes the particular playback device as a first channel of a stereo pair and an additional playback device as a second channel of the stereo pair, wherein modifying the one or more transport controls of the control interface to control playback by the particular playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the additional playback device, and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the additional playback device playing back the multimedia content as the stereo pair.

3. (Currently amended) The method of claim 1, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

additional playback device in synchrony, and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

4. (Currently amended) The method of claim 1, wherein the control interface is displayed by an application associated with the ~~content provider~~ streaming content service, and wherein the set of inputs further ~~detecting the one or more third inputs~~ comprises detecting an input to select a link in the application associated with the ~~content provider~~ streaming content service and wherein selection of the link launches a second application to facilitate retrieving ~~streaming of~~ the multimedia content [[to]] by the particular playback device from a particular source indicated by a resource locator.

5. (Currently amended) The method of claim 1, wherein the control interface is displayed by an application associated with the ~~content provider~~ streaming content service, and wherein the set of inputs further ~~detecting the one or more third inputs~~ comprises detecting an input to select a link in the application associated with the ~~content provider~~ streaming content service and wherein selection of the link causes the control device to transmit ~~provides~~ information to the one or more first cloud servers ~~a server~~ to ~~begin multimedia content playback via~~ add multimedia content to the local playback queue on the particular playback device.

6. (Currently amended) The method of claim 1, further comprising detecting, via[[by]] the control device, a set of inputs to transfer playback from the playback device back to the control device, wherein transferring playback from the playback device back to the control device comprises:

causing playback at the playback device to be stopped; and

modifying the one or more transport controls of the control interface to control playback by the control device.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

7. (Currently amended) The method of claim 1, wherein causing the graphical interface to display the control interface including one or more transport controls to control playback by the control device comprises causing the graphical interface to display a control interface that includes the one or more transport controls in a particular arrangement on the graphical interface, and wherein modifying the one or more transport controls of the control interface to control playback by the particular playback device comprises causing the graphical interface to display the one or more transport controls to control playback by the particular playback device ~~while~~ in the particular arrangement.

8. (Currently amended) The method of claim 1, wherein ~~sending information regarding the multimedia content from the control device to the playback device comprises sending, via a local area network, an identifier indicating~~ causing the one or more first cloud servers to add multimedia content to the local playback queue comprises causing an identifier of the multimedia content to be added to the local playback queue, wherein the identifier indicates a particular source of the multimedia content at the one or more second cloud servers of the streaming content service, wherein the particular playback device receives the multimedia content from the particular source at the one or more second cloud servers of the streaming content service.

9. (Currently amended) The method of claim 1, ~~wherein sending information regarding the multimedia content from the control device to the playback device comprises sending information regarding the multimedia content from the control device to a server that provides the multimedia content to the playback device~~

wherein causing one or more first cloud servers to add the multimedia content to the local playback queue on the particular playback device comprises sending a message to the streaming content service that causes the one or more first cloud servers to add the multimedia content to the local playback queue on the particular playback device.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

10. (Currently amended) A tangible, non-transitory computer readable storage medium including instructions for execution by a processor, the instructions, when executed, cause a control device to implement a method comprising:

causing a graphical interface to display a control interface including one or more transport controls to control playback by the control device;

after connecting to a local area network via a network interface, identifying playback devices connected to the local area network;

causing the graphical interface to display a selectable option for transferring playback from the control device;

detecting a set of inputs to transfer playback from the control device to a particular playback device, wherein ~~detecting~~ the set of inputs comprises: (i) a selection of the selectable option for transferring playback from the control device and (ii) a selection of the particular playback device from the identified playback devices connected to the local area network:

~~detecting a first input comprising an identification of the playback device;~~

~~detecting a second input comprising an identification of an item, wherein multimedia content associated with the item is retrievable from a content provider;~~

~~detecting a trigger, wherein detecting the trigger comprises detecting one or more third inputs that are not the first input or the second input; and~~

after detecting the set of inputs to transfer playback from the control device to the particular playback device, causing playback to be transferred from the control device to the particular playback device, wherein transferring playback from the control device to the particular playback device comprises:

(a) causing one or more first cloud servers to add multimedia content to a local playback queue on the particular playback device;

(b) causing playback at the control device to be stopped; and

(c) modifying the one or more transport controls of the control interface to control playback by the playback device; and

causing the particular playback device to play back the multimedia content, wherein the particular playback device playing back the multimedia content comprises the particular



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

playback device retrieving the multimedia content from one or more second cloud servers of a streaming content service and playing back the retrieved multimedia content

~~sending, via a network interface, information regarding the multimedia content from the control device to the playback device, wherein the information comprises an identification of the multimedia content for playback by the playback device, and wherein the information causes the playback device to (a) retrieve, independent of the control device, the multimedia content from the content provider and (b) initiate playback of the retrieved multimedia content~~.

11. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone of a media playback system that includes the particular playback device as a first channel of a stereo pair and an additional playback device as a second channel of the stereo pair, wherein modifying the one or more transport controls of the control interface to control playback by the particular playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the additional playback device, and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the additional playback device playing back the multimedia content as the stereo pair.

12. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony,



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.

13. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein the control interface is displayed by an application associated with the ~~content provider~~ streaming content service, and wherein the set of inputs further ~~detecting the one or more third inputs~~ comprises detecting an input to select a link in the application associated with the ~~content provider~~ streaming content service and wherein selection of the link launches a second application to facilitate retrieving ~~streaming of~~ the multimedia content [[to]] by the particular playback device from a particular source indicated by a resource locator.

14. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein the control interface is displayed by an application associated with the ~~content provider~~ streaming content service, and wherein the set of inputs further ~~detecting the one or more third inputs~~ comprises detecting an input to select a link in the application associated with the ~~content provider~~ streaming content service and wherein selection of the link causes the control device to transmit ~~provides~~ information to the one or more first cloud servers ~~a server~~ to ~~begin multimedia content playback via~~ add multimedia content to the local playback queue on the particular playback device.

15. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein the method further comprises ~~further comprising~~ detecting~~, by the control device,~~ a set of inputs to transfer playback from the playback device back to the control device, wherein transferring playback from the playback device back to the control device comprises:

causing playback at the playback device to be stopped; and

modifying the one or more transport controls of the control interface to control playback by the control device.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

16. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein causing the[[a]] graphical interface to display the control interface including one or more transport controls to control playback by the control device comprises causing the graphical interface to display a control interface that includes the one or more transport controls in a particular arrangement on the graphical interface, and wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the graphical interface to display the one or more transport controls to control playback by the playback device ~~while~~ in the particular arrangement.

17. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein ~~sending information regarding the multimedia content from the control device to the playback device comprises sending, via a local area network, an identifier indicating~~ causing the one or more first cloud servers to add multimedia content to the local playback queue on the particular playback device comprises causing an identifier of the multimedia content to be added to the local playback queue, wherein the identifier indicates a particular source of the multimedia content at the one or more second cloud servers of the streaming content service, wherein the particular playback device receives the multimedia content from the particular source at the one or more second cloud servers of the streaming content service.

18. (Currently amended) The tangible, non-transitory computer readable medium of claim 10,

~~wherein sending information regarding the multimedia content from the control device to the playback device comprises sending information regarding the multimedia content from the control device to a server that provides the multimedia content to the playback device~~

wherein causing one or more first cloud servers to add the multimedia content to the local playback queue on the particular playback device comprises sending a message to the streaming content service that causes the one or more first cloud servers to add the multimedia content to the local playback queue on the particular playback device.

19. (Currently amended) A control device comprising:



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

a graphical interface;

a wireless communication interface to communicate with a playback device;

one or more processors;

tangible non-transitory computer-readable media having instructions encoded therein, wherein the instructions, when executed by the one or more processors, cause the control device to perform functions comprising:

causing the graphical interface to display a control interface including one or more transport controls to control playback by the control device;

after connecting to a local area network via the wireless communication interface, identifying playback devices connected to the local area network;

causing the graphical interface to display a selectable option for transferring playback from the control device;

detecting a set of inputs to transfer playback from the control device to a particular playback device, wherein ~~detecting~~ the set of inputs comprises: (i) a selection of the selectable option for transferring playback from the control device and (ii) a selection of the particular playback device from the identified playback devices connected to the local area network:

~~detecting a first input comprising an identification of the playback device;~~

~~detecting a second input comprising an identification of an item, wherein multimedia content associated with the item is retrievable from a content provider;~~

~~detecting a trigger, wherein detecting the trigger comprises detecting one or more third inputs that are not the first input or the second input; and~~

after detecting the set of inputs to transfer playback from the control device to the particular playback device, causing playback to be transferred from the control device to the particular playback device, wherein transferring playback from the control device to the particular playback device comprises:

(a) causing one or more first cloud servers to add multimedia content to a local playback queue on the particular playback device;

(b) causing playback at the control device to be stopped; and



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

(c) modifying the one or more transport controls of the control interface to control playback by the playback device; and

causing the particular playback device to play back the multimedia content, wherein the particular playback device playing back the multimedia content comprises the particular playback device retrieving the multimedia content from one or more second cloud servers of a streaming content service and playing back the retrieved multimedia content

~~sending, via the wireless communication interface, information regarding the multimedia content from the control device to the playback device, wherein the information comprises an identification of the multimedia content for playback by the playback device, and wherein the information causes the playback device to (a) retrieve, independent of the control device, the multimedia content from the content provider and (b) initiate playback of the retrieved multimedia content~~.

20. (Currently amended) The control device of claim 19, wherein detecting the set of inputs to transfer playback from the control device to the particular playback device comprises detecting a set of inputs to transfer playback from the control device to a particular zone group of a media particular playback system that includes a first zone and a second zone, wherein the first zone includes the particular playback device and the second zone includes at least one additional playback device, wherein modifying the one or more transport controls of the control interface to control playback by the playback device comprises causing the one or more transport controls of the control interface to control playback by the particular playback device and the at least one additional playback device in synchrony, and wherein ~~initiating playback of~~ the particular playback device playing back the retrieved multimedia content comprises ~~initiating playback by~~ the particular playback device and the at least one additional playback device playing back the multimedia content in synchrony.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

21. (Currently amended) The method of claim 1, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting ~~an input that causes transfer of playback from the controller to the playback device~~ a selection of the multimedia content.

22. (Currently amended) The method of claim 1, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting an input that causes playback at the control device ~~controller~~ to be stopped.

23. (Currently amended) The method of claim 1, wherein detecting the set of inputs comprises ~~one or more third inputs~~ detecting selection of a button on the control interface.

24. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting ~~an input that causes transfer of playback from the controller to the playback device~~ a selection of the multimedia content.

25. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting an input that causes playback at the control device ~~controller~~ to be stopped.

26. (Currently amended) The tangible, non-transitory computer readable medium of claim 10, wherein detecting the set of inputs comprises ~~one or more third inputs~~ detecting selection of a button on the control interface.

27. (Currently amended) The control device of claim 19, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting ~~an input that causes transfer of playback from the controller to the playback device~~ a selection of the multimedia content.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

28. (Currently amended) The control device of claim 19, wherein detecting the set of inputs comprises ~~one or more third inputs~~ comprises detecting an input that causes playback at the control device ~~controller~~ to be stopped.

29. (Currently amended) The control device of claim 19, wherein detecting the set of inputs comprises ~~one or more third inputs~~ detecting selection of a button on the control interface.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

## REMARKS

### 1. Summary of the Office Action

In the non-final Office Action mailed July 25, 2016, the Examiner rejected claims 1, 6-10, 15-19, and 21-29 under pre-AIA 35 U.S.C. § 103(a) as being allegedly unpatentable over DaCosta (US 2008/0134256) in view of Dua (US 2006/0258289); rejected claims 3, 12, and 20 under pre-AIA 35 U.S.C. § 103(a) as being allegedly unpatentable over DaCosta in view of Dua in view of Millington (US 2012/0192071); and rejected claims 4-5 and 13-14 under pre-AIA 35 U.S.C. § 103(a) as being allegedly unpatentable over DaCosta in view of Dua in view of Zott (US 2009/0228919).

### 2. Status of the Claims

Currently pending are claims 1-29, of which claims 1, 10 and 19 are independent and the remainder are dependent. Claims 1-29 have been amended to further clarify aspects of the invention and to expedite prosecution. No new matter has been added.

### 3. Response to the § 103 Rejections

As noted above, the Examiner rejected claims 1, 9 and 20 under pre-AIA 35 U.S.C. § 103 as being allegedly unpatentable over DaCosta (US 2008/0134256) in view of Dua (US 2006/0258289). For at least the reason that the cited references do not teach the subject matter currently recited by Applicant's claims, the pending § 103 rejections should be withdrawn.

In view of the foregoing, Applicant requests that the pending art rejections of claims 1, 9 and 19 be withdrawn. In addition, the pending § 103 rejections of dependent claims 2-9, 11-18, and 20-29 should be withdrawn as well.

### 4. Conclusion

Applicant submits that claims 1-29 are in condition for allowance. Applicant does not acquiesce in any assertion by the Examiner that is not expressly addressed by these remarks. Should the Examiner wish to discuss this case with the undersigned, the Examiner is encouraged to call the undersigned at (312) 913-2128.



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002

Respectfully submitted,
**McDONNELL BOEHNEN**
**HULBERT & BERGHOFF LLP**

Date: October 25, 2016

By: /Benjamin M. Urban/
Benjamin M. Urban
Registration No. 71,591



**McDonnell, Boehnen, Hulbert & Berghoff LLP**
300 South Wacker Drive
Chicago, Illinois 60606
Telephone: (312) 913-0001
Facsimile: (312) 913-0002